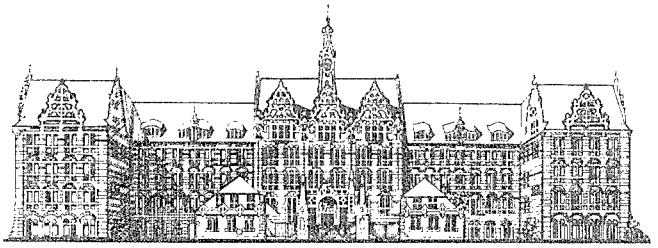
ZESZYTY NAUKOWE
WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ

TECHNOLOGIE INFORMACYJNE

TI

NR 5



1904 1945 2004/2005

JUBILEUSZ POLITECHNIKI W GDAŃSKU

BEST AVAILABLE COPY

20041206 049

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	23 August 2004	Conference Proceedings, 18 May 2004

4. TITLE AND SUBTITLE

2nd Conference on Information Technology / Special Session on Homeland Security, Volume 5

5. FUNDING NUMBERS

FA8655-04-1-5047

6. AUTHOR(S)

Conference Committee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Gdansk University of Technology
ul. Gabriela Narutowicza 11/12
80-952
Gdansk 80-952
Poland

8. Performing Organization Report Number

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES

EOARD
PSC 802 Box 14
FPO 09499-0014

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

CSP 04-5047

11. SUPPLEMENTARY NOTES

Volume 5, ISBN 83-917681-5-5

Copyright 2004 Wydzial ETI Politechniki Gdanskiej Gdansk. Available from: Wydzial ETI Politechniki Gdanskiej, Gdansk. The Department of Defense has permission to use for government purposes only. All other rights are reserved by the copyright holder.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. (approval given by local Public Affairs Office)

12b. DISTRIBUTION CODE

A

ABSTRACT (Maximum 200 words)

The Final Proceedings for 2nd Conference on Information Technology / Special Session on Homeland Security, 16-18 May 2004

Formal methods in information engineering
Electronic documents and digital libraries
Autonomous robots in embeded systems
Mobile and portable information systems
Design and implementation of methodologies and technologies for information based products
Dependability and security of information processing systems
Development of IT infrastructure
Special English Session on Homeland Security

14. SUBJECT TERMS

EOARD, Computational methods, C31, Computer network security

15. NUMBER OF PAGES

384

16. SECURITY CLASSIFICATION OF:	17. LIMITATIONS OF ABSTRACT	18a, NAME OF RESPONSIBLE PERSON
a. Report UNCLASSIFIED b. Abstract UNCLASSIFIED c. This page UNCLASSIFIED	UNCLASSIFIED	Paul Losiewicz, Ph. D. 18b. TELEPHONE NUMBER (include area code) +44 20 7514 4474

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

BEST AVAILABLE COPY

ZESZYTY NAUKOWE
WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ

TECHNOLOGIE INFORMACYJNE

NR 5

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited



Gdańsk 2004

AQ F05-02-0391

KOMITET REDAKCYJNY

Zeszytów Naukowych Wydziału ETI
Politechniki Gdańskiej

Redaktorzy wydania

Józef Woźniak
Krzysztof Nowicki

Wydano za zgodą
DZIEKANA WYDZIAŁU ETI PG

W materiałach konferencyjnych zamieszczono
wyłącznie artykuły recenzowane

© Copyright by Wydział ETI Politechniki Gdańskiej
Gdańsk 2004

ISBN 83-917681-5-5

Druk: Zakład Poligrafii Politechniki Gdańskiej
ul. G. Narutowicza 11/12, 80-952 Gdańsk, tel. (0-58) 347 25 35

Technologie informacyjne (TI) umożliwiają implementację różnego typu systemów, szeroko stosowanych we współczesnym świecie. Nie oznacza to wcale, że są to już technologie dojrzałe, a ich wprowadzenie nie wiąże się z żadnym większym ryzykiem. Ogromne tempo rozwoju TI wynika z faktu, że korzyści z ich wykorzystania bądź nadzieja na osiągnięcie znacznych korzyści w przyszłości są ciągle duże. Na ogół konwergencja (integracja wielu istniejących rozwiązań) i synergia (generowanie nowych możliwości po integracji tych rozwiązań) są motorem ciągłego postępu.

II Konferencja Technologie Informacyjne jest zorganizowana przez Wydział Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej pod patronatem trzech Prezydentów Trójmiasta: Gdańska, Gdyni, Sopotu i J. M. Rektora PG w Roku Jubileuszowym Politechniki Gdańskiej, w rocznicę 100-lecia politechniki w Gdańsku. Odbywa się także już po fakcie ponownego włączenia się Polski do rodziny Krajów Europejskich. Stąd też większy zakres tematyczny tej konferencji, jak również częściowo międzynarodowy charakter. Znacznie powiększył się także skład Komitetu Programowego oraz liczba przyjętych do druku artykułów.

Tematy konferencji skupią się wokół 13 sesji tematycznych, w tym kilka sesji jest zorganizowanych przez uczestników konferencji. Są to: Homeland Security, Nietechniczne aspekty TI, Systemy mikroelektroniczne, Systemy radiowe. Tego typu sesje specjalistyczne chcemy rozwijać w przyszłości.

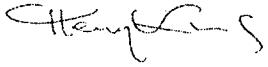
W pierwszym dniu Konferencji odbędzie się również specjalna sesja z okazji Światowego Dnia Telekomunikacji poświęcona problemom kształcenia, a wieczorem na bankiecie zostanie wręczony „Laur dla Pracodawcy" najlepszej firmie wybranej przez studentów, pracowników i absolwentów Wydziału ETI. W drugim dniu konferencji zorganizowane będą studenckie Targi Pracy. Odbędą się również dwie sesje plenarne, na których wybitni specjaliści przedstawią bardzo interesujące referaty:

1. Prof. Wojciech Szpankowski: *Ubiquitous Pattern Matching and its Applications*,
2. Prof. Józef Lubacz i prof. Andrzej Kraśniewski: *Ewolucja szkolnictwa wyższego w Europie i jej konsekwencje dla kształcenia w obszarze technik i technologii informacyjnych*.

Żałuję, że brakuje sesji poświęconej Twórcy Szkoły Mikrofal na Wydziale ETI PG Profesorowi Krzysztofowi Grabowskiemu. Niech więc wysoki poziom obrad oraz wyniki konferencji będą wyrazem podziękowania za Jego trud poświęcony rozwojowi naszego Wydziału.

Mam nadzieję, że obrady II Konferencji Technologie Informacyjne przyczynią się do lepszego wzajemnego poznania się wielu ekspertów z tej dziedziny. Wówczas dzięki skutkom konwergencji i synergii osiągniemy znacznie większy postęp we współpracy na rzecz zaspokajania potrzeb społeczeństwa informacyjnego.

Z podziękowaniem za współpracę



Dziekan Wydziału ETI
Politechnika Gdańska

Komitet Naukowy

Przewodniczący
Prof. Henryk Krawczyk (WETI PG)

Wiceprzewodniczący
Prof. Józef Woźniak (WETI PG)

Sekretarz
Dr Krzysztof Nowicki (WETI PG)

Członkowie:

Prof. Daniel Bem (WE PWr)
Prof. Michał Białko (PAN)
Prof. Zdzisław Bubnicki (PAN)
Prof. Andrzej Czyżewski (WETI PG)
Prof. Władysław Findeisen (PAN)
Prof. Tadeusz Galanc (WIiZ PWr)
Prof. Janusz Górski (WETI PG)
Prof. Tomasz Imieliński (RU USA)
Prof. Zygmunt Kitowski (AMW)
Prof. Bogdan Kosmowski (WETI PG)
Prof. Marek Kubale (WETI PG)
Prof. Krzysztof Kuchciński (LIT, SE)
Prof. Józef Lubacz (WETI PW)
Prof. Stanisław F. Łęgowski (UW, USA)
Prof. Jerzy Mazur (WETI PG)
Prof. Maciej Niedźwiecki (WETI PG)
Prof. Antoni Nowakowski (WETI PG)
Prof. Tadeusz Orzechowski (WEAliE AGH)
Prof. Zdzisław Pawlak (PAN)
Prof. Krzysztof Pawlikowski (CU, NZ)
Prof. Michał Polowczyk (WETI PG)
Prof. Andrzej Ruciński (UNH, USA)
Prof. Dominik Rutkowski (WETI PG)
Prof. Jerzy Rutkowski (WAEil PŚl.)
Prof. Roman Salamon (WETI PG)
Prof. Wojciech Sobczak (WETI PG)
Prof. Andrzej Stepnowski (WETI PG)
Prof. Wojciech Szpankowski (PU, USA)
Prof. Jan Węglarz (PAN)
Prof. Stefan Węgrzyn (PAN)
Prof. Bogdan M. Wilamowski (AU, USA)
Prof. Bogdan Wiszniewski (WETI PG)
Prof. Wiesław Woliński (PAN)
Prof. Romuald Zielonko (WETI PG)
Prof. Marian Zientalski (WETI PG)
Prof. Jacek M. Żurada (UL, USA)

Recenzenci

Andrzej Czyżewski
Krzysztof Goczyła
Ewa Hermanowicz
Wojciech Jędruch
Jerzy Kaczmarek
Sylwester Kaczmarek
Renata Kalicka
Ryszard Katulski
Bogdan Kosmowski
Bożena Kostek
Henryk Krawczyk
Marek Kubale
Jerzy Mazur
Maciej Niedźwiecki
Antoni Nowakowski
Michał Polowczyk
Andrzej Ruciński
Dominik Rutkowski
Roman Rykaczewski
Wojciech Sobczak
Bogdan Wiszniewski
Józef Woźniak
Romuald Zielonko
Marian Zientalski

SPIS TREŚCI

MODUŁ GENEROWANIA PROSTYCH KOMUNIKATÓW MODALNYCH
Radosław P. Katarzyniak, Rafał Owczarek 661

TRANSFORMACJE MODELU ARCHITEKTONICZNEGO OPROGRAMOWANIA
Alina Kmiecik, Vincenzo Ambriola 671

PODEJŚCIE DO JAKOŚCI W METODOLOGIACH WYTWARZANIA OPROGRAMOWANIA
Marek Kostrzewa 679

INTEGRACJA MODELOWANIA I IMPLEMENTACJI OPROGRAMOWANIA W METODZIE IMAC
Jarosław Kuchta 687

HISTORIA REZULTATÓW TESTÓW JEDNOSTKOWYCH TWORZONA ZA POMOCĄ CRUISECONTROL I MAVEN
Jerzy Nawrocki, Piotr Pawałowski, Marcin Werla 699

BADANIE WPŁYWU KONSTRUKCJI MODELU WIELOAGENTOWEGO NA JEGO ZDOLNOŚĆ PREDYKCYJNĄ
Rafał Sworowski, Henryk Lasota 707

ALOKACJA ZADAŃ W SYSTEMIE RÓWNOLEGLE PRACUJĄCYCH MASZYN
Zbigniew Buchalski 717

ANALIZA PRZYBLIŻONEGO ALGORYTMU DLA PROBLEMU SZUKANIA DRZEWA SPINAJĄCEGO O MINIMALNYM UPORZĄDKOWANYM INDEKSIE CHROMATYCZNYM
Dariusz Dereniowski 725

STRATEGIE ALGORYTMICZNE RYWALIZACJI AUTONOMICZNYCH OBIEKTÓW MILITARNYCH NA PLATFORMIE DEDYKOWANEJ ROBOCODE
Marcin Fuszara, Adrian Kosowski 733

UOGÓLNIONE ALGORYTMY ZACHŁANNE W KONTRASTOWYM KOLOROWANIU GRAFÓW
Robert Janczewski 743

RÓWNOLEGŁE ALGORYTMY GENETYCZNE W ZASTOSOWANIU DO PROBLEMU KOLOROWANIA WIERZCHOŁKÓW GRAFU
Krzysztof Kwarciany, Zbigniew Kokosiński 747

KONCEPCJA ŚRODOWISKA UMOŻLIWIAJĄCEGO SYMULACJĘ DZIAŁAŃ OBIEKTÓW MILITARNYCH
Michał Małafiejski, Adrian Kosowski, Marcin Fuszara 755

PAKIET WSPOMAGANIA TWORZENIA I OPTYMALIZACJI HARMONOGRAMÓW ZAJĘĆ AKADEMICKICH
Stefan Pepol, Konrad Piwakowski 765

PORÓWNANIE WYBRANYCH PROGRAMÓW KOMPUTEROWYCH DO UKŁADANIA ROZKŁADÓW ZAJĘĆ
Tomasz Witkowski 777

STANOWISKO LABORATORYJNE DO TESTOWANIA ANALOGOWYCH UKŁADÓW ELEKTRONICZNYCH Z WYKORZYSTANIEM MAGISTRALI IEEE 1149.4
Bogdan Bartosiński 785

ANALOGOWE FILTRY PASMA PODSTAWOWEGO SCALONYCH ODBIORNIKÓW TELEFONII KOMÓRKOWEJ
Jacek Jakusz, Waldemar Jendernalik 793

STATYSTYCZNA ANALIZA PROCESORÓW ASYNCHRONICZNYCH
Wojciech Kadłubowski 799

IMPLEMENTACJA WYBRANYCH STRUKTUR SZTUCZNYCH SIECI NEURONOWYCH W CYFROWYCH UKŁADACH PROGRAMOWALNYCH
Miron Kłosowski, Sławomir Malinowski 807

NOWY, DWUDRENOWY TRANZYSTOR MOS JAKO CZUJNIK POLA MAGNETYCZNEGO
Wiesław Kordalski, Michał Polowczyk, Janusz Czerwiński, Jerzy Dzierżko, Lech Dobrzański 815

ADMITANCJA WEJŚCIOWA I TRANSADMITANCJA MOSFETA W ZAKRESIE B.W.CZ.
Wiesław Kordalski, Tomasz Stefański 825

CONTINUOUS-TIME ACTIVE-RC FILTERS: GENERAL MODEL AND ITS APPLICATIONS
Slawomir Koziel 833

NONLINEAR DISTORTION ANALYSIS AND OPTIMIZATION OF GENERAL OTA-C FILTERS
Slawomir Koziel, Stanislaw Szczepanski 843

REZYSTOR AKTYWNY CMOS ORAZ JEGO ZASTOSOWANIE DO BUDOWY WZMACNIACZA TRANSKONDUKTANCYJNEGO
Bogdan Pankiewicz, Mariusz Solecki 851
ALGORYTMICZNE METODY REDUKCJI POBORU MOCY W CYFROWYCH UKŁADACH CMOS
Władysław Szcześniak, Piotr Szcześniak 859
NOWE STRUKTURY KORYGOWANYCH LASEREM REZYSTORÓW WARSTWOWYCH ZAPEWNIAJĄCE POPRAWĘ DOKŁADNOŚCI STROJENIA I STABILNOŚCI
Marek Wroński 865
NOWE PODEJŚCIE DO PROJEKTOWANIA I REALIZACJI KOREKCJI LASEROWEJ REZYSTORÓW WARSTWOWYCH
Marek Wroński 875
TRÓJWYMIAROWA WIZUALIZACJA OBIEKTÓW NA DNIE MORSKIM Z ZASTOSOWANIEM TECHNIK WIRTUALNEJ RZECZYWISTOŚCI
Krzysztof Bikonis, Marek Moszyński, Andrzej Stepnowski 883
ŚREDNIOWIECZNE RĘKOPISY W ZASOBACH ŚWIATOWEGO INTERNETU
Aldona Chlewicka 891
WYKORZYSTANIE JĘZYKA RDF DO OPISU I PRZESZUKIWANIA ZBIORU TEKSTÓW LITERACKICH
Michał Franczak, Krzysztof Juszczyszyn 899
ZASTOSOWANIE RELACYJNYCH BAZ DANYCH I LOGIKI OPISOWEJ W SYSTEMACH INTEGRACJI DANYCH
Teresa Grabowska, Michał Zawadzki 907
OKREŚLANIE PODOBIEŃSTWA KSZTAŁTU OBIEKTU DO LITERY LUB CYFRY
Jacek Lebiedź 915
POMIAR WYDAJNOŚCI SYSTEMÓW GRAFIKI TRÓJWYMIAROWEJ
Jacek Lebiedź, Marcin Stankiewicz 923
WIELOWYMIAROWA WIZUALIZACJA INFORMACJI
Jacek Lebiedź, Marta Stępniak 931
ONTOLOGICZNE MODELOWANIE RZECZYWISTOŚCI W BIBLIOTEKACH CYFROWYCH
Marek Nahotko 941
ZASTOSOWANIE ZACHOWUJĄCEJ KOLOR SEGMENTACJI OBRAZU BARWNEGO DO WYSZUKIWANIA OBSZARÓW ZAWIERAJĄCYCH TEKST
Marcin Pazio 949
DWUPOZIOMOWE CYFROWE ZNAKOWANIE OBRAZÓW NIERUCHOMYCH
Roman Rykaczewski 957
EKSTRAKCJA ORAZ DOPASOWYWANIE KONTURÓW MELODYCZNYCH W BAZACH DANYCH MUZYCZNYCH
Wojciech Seliga, Mariusz Szwoch 965
BADANIE WEKTORA PARAMETRÓW DO AUTOMATYCZNEGO ROZPOZNAWANIA STYLÓW MUZYCZNYCH
Piotr Szczuko, Bożena Kostek 975
WYSZUKIWANIE OBRAZÓW W MULTIMEDIALNYCH BAZACH DANYCH
Wioleta Szwoch 983
PRZETWARZANIE WSTĘPNE I SEGMENTACJA ZNISZCZONYCH DOKUMENTÓW ARCHIWALNYCH
Mariusz Szwoch, Wioleta Szwoch 993
WYWODZENIE SCHEMATU Z DANYCH O NIEZNANEJ STRUKTURZE ZA POMOCĄ ALGORYTMU COBWEB S+T
Wojciech Waloszek 1003
PROTOKOLLANALYSE BEI VoIP-SYSTEMEN
Krzysztof Nowicki, Tadeus Uhl 1011
AKWIZYCJA I PRZETWARZANIE DANYCH W SYSTEMIE TELEMONITORINGU SIECI CIEPŁOWNICZEJ
Lech Hasse, Marek Illeczko 1023
ZASTOSOWANIE OPROGRAMOWANIA COMSIS DO OCENY DYNAMICZNEJ ODPORNOŚCI NA ZAKŁÓCENIA CYFROWYCH UKŁADÓW KOMBINACYJNYCH
Lech Hasse, Józef Piasecki, Ludwik Spiralski 1029
KSZTAŁCENIE INŻYNIERÓW ICT W ŚWIECIE PERMANENTNYCH ZMIAN
Henryk Lasota 1035

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Radosław P. Katarzyniak, Rafał Owczarek

Instytut Sterowania i Techniki Systemów, Politechnika Wrocławska

MODUŁ GENEROWANIA PROSTYCH KOMUNIKATÓW MODALNYCH

Streszczenie

Przedstawiono model organizacji modułu odpowiedzialnego za generowanie prostych komunikatów modalnych przez agenta kognitywnego. Założono, że komunikaty są modalnymi rozszerzeniami prostych wypowiedzi agenta na temat stanu cechy w określonym obiekcie świata zewnętrznego. Rozpatrzono użycie modalnych operatorów wiedzy, przekonań i możliwości. Przyjęto wewnętrzną organizację agenta realizującą wybrane aspekty organizacji umysłu człowieka. Zdefiniowano tzw. relację gruntowania wypowiedzi prostych ustalającą warunki zgodności intencjonalnej treści generowanej wypowiedzi ze stanem baz wiedzy empirycznej zgromadzonej przez agenta. Podano przykładowy algorytm implementujący zaproponowaną relację. W algorytmie wykorzystano elementarne mechanizmy statystyczne.

1. WSTĘP

Zagadnienie generowania języków semantycznych tj. języków posiadających określone znaczenie intencjonalne, jest jednym z podstawowych problemów sztucznej inteligencji, robotyki i nauk kognitywnych. Celem wymienionych dziedzin jest w szczególności budowanie modeli sztucznych jednostek autonomicznych (agentów, robotów) posiadających umiejętność generowania komunikatów o określonym znaczeniu (treści). Problem generowania wyrażeń semantycznego języka komunikacji jest ściśle związany ze zjawiskiem gruntowania wyrażeń w strukturach wiedzy „umysłu" przechowujących treści poznawcze dostępne na poziomie mentalnym [1]. Jak zakłada lingwistyka kognitywna [2] oraz fenomenologia [3] [4] pierwotnym źródłem każdej treści dotyczącej świata zewnętrznego jest doświadczenie empiryczne agenta zdobywane na drodze bezpośredniej interakcji z otoczeniem. Stąd problem generowania języków semantycznych u swoich podstaw sprowadza się (w zakresie dotyczącym treści generowanych komunikatów) do zadania zdefiniowania relacji łączącej komunikat z całokształtem wiedzy ucieleśniającej zebrane wcześniej doświadczenia oraz do ustalenia kryteriów wyboru komunikatów najlepiej ugruntowanych w tych doświadczeniach (najlepiej reprezentujących zgromadzony stan wiedzy empirycznej).

Zasygnalizowany wyżej związek pomiędzy językiem semantycznym a zawartością baz wiedzy przechowujących zebrane doświadczenia empiryczne w oryginalny sposób badany był w kilku pracach, w których rozpatrzono wybrane przypadki gruntowania prostych zinterpretowanych logicznych języków semantycznych, w szczególności języka formuł atomowych i modalnych rozszerzeń formuł atomowych [5], języka formuł wiedzy ze spójnikiem alternatywy i alternatywy wykluczającej [6] [7] [8], języka formuł przekonań ze spójnikiem koniunkcji, alternatywy i alternatywy wykluczającej [7] [8] [9] oraz języka formuł wiedzy, przekonań i możliwości ze spójnikiem implikacji [10] [11].

W innych pracach pokazano, w jaki sposób zaproponowane modele gruntowania można wykorzystać w obszarze wspomagania decyzji np. [12].

W każdym z wymienionych wyżej tekstów omawiających gruntowanie wybranego języka semantycznego oraz definiującego kryteria wyboru najlepiej ugruntowanego komunikatu (wypowiedzi) przyjmowano jako obowiązujący określony model wewnętrznych baz wiedzy agenta. Modele prezentowane w poszczególnych pracach w różnym stopniu odwoływały się do założeń dotyczących organizacji umysłu i roli tej organizacji w komunikacji zakładanej przez lingwistykę i nauki kognitywne. W szczególności, w modelach prezentowanych w pracach [5] [6] [7] [8] [9] [10] nie przyjęto podziału wewnętrznych procesów przetwarzania wiedzy i generowania języka na procesy „świadome" (ukierunkowane werbalnie) i „nieświadome" (ukierunkowane przed werbalnie) [13]. Odmienne podejście zaprezentowano w pracy [11], gdzie całość materiału empirycznego stanowiącego źródło znaczenia komunikatu modalnego ze spójnikiem implikacji podzielono na dwa obszary reprezentujące wymieniony wyżej obszar „świadomy" i „nieświadomy". W ostatnim przypadku tj. [11] uwzględnienie podziału było konieczne, gdyż w systemach żywych on właśnie stanowi źródło niepewności wiedzy uzasadniającej i wymagającej użycie spójnika implikacji w połączeniu z modalnymi operatorami możliwości i przekonań.

W niniejszej pracy dwupoziomowy model organizacji bazy wiedzy empirycznej wykorzystywany zastosowano także dla przypadku agenta generującego modalne wypowiedzi proste. Ponadto dla podanej organizacji dwupoziomowej zaproponowano przykładowy sposób implementacji relacji gruntowania odwołujący się do statystycznej aproksymacji rozkładu doświadczeń empirycznych w „świadomym" i „nieświadomym" obszarze „umysłu" agenta. Przyjęto, że algorytm ten zgodnie z założeniami architektury dwumodułowej (dwupoziomowej), najpierw przeszukiwać będzie obszar „świadomy", a następnie odwoływać będzie się do obszaru „nieświadomego". Źródłem niepewności sygnalizowanej przez agenta za pomocą użycia modalnego operatora możliwości i przekonań są ograniczenia w pełnym dostępie procedur wyboru komunikatu do obszaru „nieświadomego" (zazwyczaj przechowującego większość materiału empirycznego kształtującego znaczenie konkretnego komunikatu). Obszar „świadomy" jest w tym ujęciu odpowiednikiem obszaru pamięci roboczej, który w naturalnych agentach kognitywnych zawsze posiada pojemność mniejszą od pojemności pozostałej części „umysłu".

2. MODEL AGENTA

W pracy zakłada się, że kognitywny agent znajduje się w określonym świecie rzeczywistym *W*. W świecie tym istnieją wyłącznie obiekty atomowe. Obiekty te zaopatrzone są w zestaw charakteryzujących je cech. Świat *W* postrzegany jest przez agenta jako system dynamiczny, w którym zmianom podlegają jedynie stany cech poszczególnych obiektów atomowych, natomiast zbiór obiektów jest stały. Przeprowadzając obserwacje agent może stwierdzić, czy konkretny obiekt atomowy w momencie przeprowadzania obserwacji

$t_k \in T=\{t_0,t_1,...\}$ posiada określoną cechę. Percepcja towarzysząca obserwacji ograniczona jest zazwyczaj do fragmentu otoczenia wyróżnionego przez pole uwagi agenta. W przypadku, gdy pewien obiekt znajduje się poza polem percepcji agenta, obiekt ten uznany musi być za pozostający w obszarze tzw. niewiedzy agenta zrelatywizowanej do punktu czasowego t_k.

Treść pojedynczej percepcji rejestrowana jest w „umyśle" w postaci złożonej struktury formalnej zwanej profilem bazowym np. [14] [15]:

Definicja. *Reprezentacja treści pojedynczej percepcji otoczenia przeprowadzonej przez agenta w punkcie czasowym $t_k \in T=\{t_0,t_1,...\}$ dana jest jako system relacyjny następującej postaci:*

$$BP(t_k)=<O,P^+_1(t_k),P^-_1(t_k),...,P^+_K(t_k),P^-_K(t_k)> \quad (2.1)$$

gdzie:

1. Zbiór $O = \{o_1,o_2,...,o_M\}$ zawiera obiekty atomowe występujące w świecie *W*.
2. Każda para $P^+_i(t_k) \subseteq O$, $P^-_i(t_k) \subseteq O$ (i=1,2,..., K) opisuje stan cechy P_i w zaobserwowanych w punkcie t_k obiektach atomowych ze zbioru *O*. Dla każdego obiektu $o \in O$ relacja $o \in P^+_i(t_k)$ zachodzi wtedy i tylko wtedy, gdy obserwując w punkcie czasowym t_k obiekt *o* agent stwierdził wystąpienie w nim cechy P_i oraz $o \in P^-_i(t_k)$ wtedy i tylko wtedy, gdy obserwując w punkcie czasowym t_k obiekt *o* agent stwierdził brak wystąpienia cechy P_i w obiekcie *o*.
3. Dla każdej pary zbiorów $P^+_i(t_k) \subseteq O$, $P^-_i(t_k) \subseteq O$ (i=1,2,..., K) zachodzi $P^+_i(t_k) \cap P^-_i(t_k) = \varnothing$.

Zbiór wszystkich zapamiętanych percepcji stanowi całkowitą bazę wiedzy empirycznej agenta na temat możliwych stanów otoczenia i wyznacza faktyczny stan jego wiedzy w tym zakresie. Formalnie bazę tę przedstawić można jako kolekcję profili bazowych uporządkowaną według czasu:

Definicja. *Stan wiedzy agenta w punkcie czasowym t_b dany jest jako kolekcja profili bazowych i przechowywany jest w Temporalnej Bazie Wiedzy:*

$$KS\ (t_b) = \{ BP\ (t) : t \in \{ 0,1,2 \ldots t_b \} \} \quad (2.2)$$

W celu uwzględnienia wspomnianego wyżej założenia o dwupoziomowości reprezentacji wiedzy lingwistycznej [11] [13] przyjmuje się, że w każdym momencie t_k baza wiedzy $KS(t_k)$ podzielona jest na dwa zbiory zwane obszarem przetwarzania „świadomego" $KSE(t_k)$ oraz obszarem przetwarzania „nieświadomego" $KSU(t_k)$. Obszar $KSE(t_k)$ zawiera wiedzę, która w momencie t_k jest przetwarzana przez agenta na poziomie procesów obliczeniowych ulokowanych w pamięci roboczej $KSE(t_k)$ bezpośrednio dostępnej dla procesów decyzyjnych ustalających poprawność ugruntowania poszczególnych komunikatów języka (tabela 3.1). Natomiast obszar $KSU(t_k)$ to część wiedzy nieprzetworzona przez „świadome" procesy obliczeniowe, a więc wiedza wpływająca na wybór najlepszego komunikatu, jednak na zasadach analogicznych do wpływu poziomu „nieświadomego" (przedwerbalnego" [13]).

W naturalnych agentach istnienie obszaru $KSU(t_k)$ sygnalizowane jest na poziomie „świadomym" w ten sposób, iż agent nie poznaje dokładnej treści materiału empirycznego

ulokowanego w $KSU(t_k)$ i kształtującego znaczenia komunikatów z tabeli 3.1, jednak przyjmuje do wiadomości jego istnienie. Oczywiście, materiał ten może zostać przemieszczony do „świadomego" obszaru roboczego na drodze przypomnienia, prowadzi to jednak do wypchnięcia z obszaru roboczego innych elementów jego dotychczasowej zawartości.

W technicznym ujęciu wymienione obszary przedstawić można jako dwa wydzielone obiekty (moduły) stanowiące komplementarne części „umysłu" agenta. Obiekty te, aczkolwiek współpracujące w podejmowaniu decyzji o wyborze komunikatów, posiadają obszar ściśle prywatny, do którego można sięgać wyłącznie za pośrednictwem wyspecjalizowanych interfejsów. Taka modułowość i enkapsulacja modułów „umysłu" jest raczej naturalnym zjawiskiem w systemach żywych i posiada istotny wpływ na procesy generowania języka (patrz ponownie werbalny i przedwerbalny poziom reprezentacji pojęć [13]). Z praktycznego punktu widzenia stwierdzić należy, że pomiędzy obszarami $KSE(t_k)$ i $KSU(t_k)$ zachodzi specyficzna relacja wzajemnego oddziaływania prowadząca do następującego porządku:

1. Jeżeli w sytuacji uniemożliwiającej przeprowadzenie bezpośredniej obserwacji zewnętrznej agent stara się ustalić stan cechy P w obiekcie o, wtedy obszar $KSE(t_k)$ dąży do poznania sumarycznego (wypadkowego) doświadczenia stanu tej cechy w obiekcie świata, zapamiętanego w obu obszarach $KSE(t_k)$ i $KSU(t_k)$ oraz rzutowania (projekcji) tego doświadczenia na bieżące wyobrażenie stanu cechy P w obiekcie o.
2. Pełny dostęp obszaru $KSE(t_k)$ do obszaru $KSU(t_k)$ nie jest możliwy. Stąd na poziomie $KSE(t_k)$ agent ma możliwość posługiwać się co najwyżej przybliżeniem zawartości obszaru $KSU(t_k)$.
3. W celu wyznaczenia wspomnianego przybliżenia zawartości $KSU(t_k)$ obszar $KSE(t_k)$ kieruje do obszaru $KSU(t_k)$ za pośrednictwem interfejsu zapytanie o stan interesującej go cechy. Prowadzi to na poziomie obszaru $KSU(t_k)$ do uruchomienia enkapsulowanych procedur wyznaczania zbioru doświadczeń związanych z cechą P w obiekcie o. W tym właśnie sensie obszar $KSE(t_k)$ eksploruje obszar $KSU(t_k)$.

W niniejszej pracy przyjmuje się, że enkapsulowana w $KSU(t_k)$ metoda, która wywoływana jest przez obszar $KSE(t_k)$ w celu ustalenia siły doświadczeń empirycznych relewantnych do znaczeń komunikatów z tabeli 3.1, jest metodą statystyczną. Jej zarys podany zostanie w dalszej części pracy.

3. JĘZYK KOMUNIKACJI I KLASYFIKACJA WIEDZY EMPIRYCZNEJ

Jak zaznaczono wyżej język komunikacji semantycznej wykorzystywany przez rozpatrywanego agenta podano w tablicy 3.1. Na język ten składają się komunikaty zbudowane z modalnych operatorów możliwości, przekonań i wiedzy, których argumentem są proste stwierdzenia $P(o)$ o stanie cechy P w obiekcie o. Z koniecznością wyboru i wygenerowania komunikatów wymienionych w tabeli agent spotyka się udzielając odpowiedzi np. na pytanie proste postaci „Czy obiekt o posiada cechę P?".

Wybór komunikatu języka najlepiej ugruntowanego w zgromadzonym doświadczeniu empirycznym (dla punktu czasowego t_k) wymaga przeprowadzenia analizy zawartości obszarów $KSE(t_k)$ i $KSU(t_k)$. Celem takiej analizy jest ustalenie siły doświadczenia empirycznego uczestniczącego w kształtowaniu intencjonalnego znaczenia wyrażenia $P(o)$ oraz

rozkładu tego doświadczenia w obszarach $KSE(t_k)$ i $KSU(t_k)$. Należy zauważyć, że znaczenie komunikatu $P(o)$ skonstruowane przez procesy poznawcze kognitywnego agenta jest pochodną treści profili bazowych kodujących obserwacje, w których obiekt o zaobserwowano jako posiadający albo nie posiadający cechy P. Obserwacje te określane będą dalej jako *relewantny materiał empiryczny*. Z punktu widzenia lingwistyki kognitywnej pozostałe obserwacje wydają się być obojętne dla procesu utrwalania znaczenia komunikatu $P(o)$ i jego modalnych rozszerzeń.

Tablica 3.1

Język komunikacji

Lp.	Komunikat	Znaczenie komunikatu
1	Pos($P(o)$)	Możliwe, że obiekt o posiada cechę P.
2	Bel($P(o)$)	Sądzę, że obiekt o posiada cechę P.
3	Know($P(o)$)	Wiem, że obiekt o posiada cechę P.

Dla każdego ustalonego punktu czasowego t_k sumaryczną siłę obserwacji relewantnych uczestniczących w kształtowaniu znaczenia $P(o)$ opisać można następującymi zbiorami punktów czasowych:

$$Ev^{+}(P,t_k,o)= \{t: t\in \{t_0,t_1,t_2,..,t_{k-1}\} \text{ i } o\in P^{+}(t)\} \tag{3.1}$$

$$Ev^{-}(P,t_k,o)= \{t: t\in \{t_0,t_1,t_2,..,t_{k-1}\} \text{ i } o\in P^{-}_{i}(t)\} \tag{3.2}$$

W związku z tym, iż w niniejszej pracy przyjmuje się dodatkowe wymiary wewnętrznej organizacji agenta reprezentowane przez zbiory $KSE(t_k)$ i $KSU(t_k)$, zawartość zbiorów $Ev^{+}(P,t_k,o)$ oraz $Ev^{-}(P_i, t_k,o)$ należy dodatkowo opisać klasyfikując ich elementy do obszaru „świadomego" lub „nieświadomego" ustalonego w punkcie czasowym t_k. W tym celu określa się następującą dodatkową klasyfikację materiału relewantnego:

$$EEv^{+}(P,t_k,o)=Ev^{+}(P,t_k,o)\cap\{t: t\in \{t_0,t_1,t_2,..,t_{k-1}\} \text{ i } BP(t)\in KSE(t_k)\} \tag{3.3}$$

$$UEv^{+}(P,t_k,o)=Ev^{+}(P,t_k,o)\cap\{t: t\in \{t_0,t_1,t_2,..,t_{k-1}\} \text{ i } BP(t)\in KSU(t_k)\} \tag{3.4}$$

$$EEv^{-}(P,t_k,o)=Ev^{-}(P,t_k,o)\cap\{t: t\in \{t_0,t_1,t_2,..,t_{k-1}\} \text{ i } BP(t)\in KSE(t_k)\} \tag{3.5}$$

$$UEv^{-}(P,t_k,o)=Ev^{-}(P,t_k,o)\cap\{t: t\in \{t_0,t_1,t_2,..,t_{k-1}\} \text{ i } BP(t)\in KSU(t_k)\} \tag{3.6}$$

4. ZASADY GRUNTOWANIA KOMUNIKATÓW

Podana klasyfikacja relewantnego materiału empirycznego pozwala zdefiniować warunki dobrego ugruntowania poszczególnych komunikatów i, w tym sensie, ustalić kryteria pozwalające agentowi na wybór najlepszej lingwistycznej reprezentacji wiedzy o stanie cechy P w obiekcie o.

4.1. Gruntowanie operatora możliwości

Niech dany będzie stan wiedzy $KS(t)$. Dla każdej formuły $Pos(P(o))$ relacja gruntowania $KS(t)|=_G Pos(P(o))$ zachodzi wtedy i tylko wtedy, gdy:

1. Stan cechy P w obiekcie o nie jest znany dla robota w momencie t tj. $o \in P^{\pm}(t)$.
2. W przeszłości agent co najmniej jeden raz zaobserwował sytuację, w której obiekt o posiadał cechę P tj.

$$UEv^{+}(P,t,o) \cup EEv^{+}(P,t,o) \neq \emptyset \tag{4.1}$$

3. Istnieje materiał relewantny ulokowany w obszarze „nieświadomym"

$$UEv^{+}(P,t,o) \cup UEv^{-}(P,t,o) \neq \emptyset \tag{4.2}$$

4. Udział doświadczeń empirycznych potwierdzających faktyczne występowanie cechy P w obiekcie o w całym materiale relewantnym nie przekracza poziomu uzasadniającego użycie operatora możliwości:

$$0 < \frac{card\,(UEv^{+}(P,t,o) \cup EEv^{+}(P,t,o))}{card\,(UEv^{+}(P,t,o) \cup UEv^{-}(P,t,o) \cup EEv^{+}(P,t,o) \cup EEv^{-}(P,t,o))} \leq \lambda^{P} \tag{4.3}$$

Użyty powyżej symbol $X|=_G Y$ reprezentuje stwierdzenie *Stan wiedzy X jest dobrze opisany przez formułę Y* (*Y jest dobrze ugruntowane w X*), natomiast λ^P jest wartością progu uzasadniającego i dopuszczającego użycie modalnego operatora możliwości jako uzupełnienia komunikatu podstawowego danego w postaci $P(o)$.

Należy podkreślić, iż z punktu widzenia procesów decyzyjnych realizowanych w obszarze „świadomym" $KSE(t)$ nie jest możliwe dokładne wyznaczenie wartości ilorazu wymienionego w punkcie 4 podanej definicji gruntowania. Procesy te nie mają bowiem bezpośredniego dostępu do zawartości empirycznej ulokowanej w obszarze $KSU(t)$, w szczególności do obszarów UEv^{+} i UEv^{-}. Fakt ten musi być uwzględniony w każdej implementacji agenta korzystającego z podanej definicji gruntowania modalnego operatora możliwości.

4.2. Gruntowanie operatora przekonań

Niech dany będzie stan wiedzy $KS(t)$. Dla każdej formuły $Bel(P(o))$ relacja gruntowania $KS(t)|=_G Bel(P(o))$ zachodzi wtedy i tylko wtedy, gdy:

1. Spełnione są warunki 1-3 definicji gruntowania $Pos(P(o))$ podanej w p. 4.1.
2. Udział doświadczeń empirycznych potwierdzających faktyczne występowanie cechy P w obiekcie o w całym materiale relewantnym nie przekracza poziomu uzasadniającego użycie operatora możliwości:

$$\lambda^{P} < \frac{card\,(UEv^{+}(P,t,o) \cup EEv^{+}(P,t,o))}{card\,(UEv^{+}(P,t,o) \cup UEv^{-}(P,t,o) \cup EEv^{+}(P,t,o) \cup EEv^{-}(P,t,o))} \leq \lambda^{B} \tag{4.4}$$

Podobnie jak dla przypadku modalnego operatora możliwości, także dla modalnego operatora przekonań decyzję o użyciu tego ostatniego należy podjąć uwzględniając istnienie podziału na obszar „świadomy" i „nieświadomy". Symbol λ^B oznacza maksymalny próg intensywności doświadczeń przyjęty przez agenta jako obligujący go do użycia komunikatu $P(o)$ rozszerzonego przez modalny operator przekonań.

4.3. Gruntowanie operatora wiedzy

Niech dany będzie stan wiedzy $KS(t)$. Dla każdej formuły $Know(P(o))$ relacja $KS(t)|=_G Know(P(o))$ zachodzi wtedy i tylko wtedy, gdy stan cechy P w obiekcie o jest znany z obserwacji w momencie t i jest on pozytywny w zakresie cechy P tj. $o \in P^+(t)$.

Podana sytuacja nie wymaga odwoływania się do podziału materiału empirycznego na obszary $KSE(t)$ i $KSU(t)$. Możliwość użycia modalnego operatora wiedzy wynika bowiem wprost z bezpośredniego ugruntowania komunikatu $P(o)$ w ostatniej zrealizowanej obserwacji.

5. ALGORYTM WYBORU PROSTEGO KOMUNIKATU MODALNEGO

Przedstawione definicje ustalają specyficzną strategię decyzyjną, której celem jest zweryfikowanie stopnia ugruntowania komunikatów z tabeli 3.1 w materiale empirycznym zgromadzonym przez agenta. Poniższy algorytm realizujący tę strategię jest kilkuetapowy – decyzja o wyborze komunikatu podejmowana jest bowiem przez proces przebiegający w obszarze $KSE(t)$, jednak odwołujący się także do procesów przetwarzania wiedzy przebiegających w obszarze $KSU(t)$. Kroki algorytmu są następujące:

Krok 1. Ustalenie przez agenta zawartości jego wiedzy na temat stanu cechy P w obiekcie o, zakodowanej w ostatniej obserwacji (w najnowszym profilu bazowym) $BP(t_k)$. Jeżeli dla ostatniej obserwacji $BP(t_k)$ zachodzi $o \in P^+(t_k)$ tj. stan cechy P jest ściśle ustalony na wystąpienie cechy, wtedy zastosowanie ma definicja podana w p. 4.3. Zachodzi $KS(t_k)|=_G Know(P(o))$. Agent generuje komunikat $Know(P(o))$. W przeciwnym przypadku następuje przejście do **Krok 2**.

Krok 2. Jeżeli zachodzi $o \in P^{\pm}(t_k)$, konieczne staje się ustalenie czy zachodzą warunki 2-3 podane w definicjach gruntowania $KS(t_k)|=_G Pos(P(o))$ oraz $KS(t_k)|=_G Bel(P(o))$. Jeżeli warunki te zachodzą, następuje przejście do **Krok 3**, w przeciwnym przypadku do **Krok 4**.

Krok 3. Następuje oszacowanie wartości

$$\lambda = \frac{card(UEv^+(P,t_k,o) \cup EEv^+(P,t_k,o))}{card(UEv^+(P,t_k,o) \cup UEv^-(P,t_k,o) \cup EEv^+(P,t_k,o) \cup EEv^-(P,t_k,o))} = \frac{card(UEv^+(P,t_k,o)) + card(EEv^+(P,t_k,o))}{card(UEv^+(P,t_k,o)) + card(UEv^-(P,t_k,o)) + card(EEv^+(P,t_k,o)) + card(EEv^-(P,t_k,o))} \quad (5.1)$$

W tym celu realizowana jest następująca procedura szczegółowa:
Ustalany jest wolumen $Vol(t_k)$ profili relewantnych dostępnych w obszarze $KSU(t_k)$ tj. $Vol(t_k) = card(UEv^+(P,t_k,o) \cup UEv^-(P,t_k,o)) = card(UEv^+(P,t_k,o)) + card(UEv^-(P,t_k,o))$.

W obliczeniu wartości Vol(t_k) uczestniczą procesy enkapsulowane w obszarach $KSU(t_k)$ i $KSE(t_k)$. Mianowicie, proces decyzyjny enkapsulowany na poziomie $KSE(t_k)$ skierowuje za pośrednictwem istniejącego interfejsu zapytanie do poziomu $KSU(t_k)$. Poziom $KSU(t_k)$ zwraca jedynie wartość Vol(t_k), ponieważ jak założono dokładna analiza udziału $UEv^+(P,t_k,o)$ oraz $UEv^-(P,t_k,o)$ przez procesy z obszaru $KSE(t_k)$ nie jest możliwa.

Z obszaru $UEv^+(P,t_k,o) \cup UEv^-(P,t_k,o)$ losowanych jest J prób o długości L zawierających profile relewantne zapamiętane w obszarze $KSU(t_k)$.

Dla każdej próby i=1,2,....J ustalana jest statystyka próby $p_i = (n_{i,1}, n_{i,2})$, w której symbol $n_{i,1}$ oznacza licznik wystąpienia, natomiast $n_{i,2}$ licznik braku wystąpienia cechy P w wylosowanych profilach. Oczywiście, dla każdego i = 1,2,...,J, zachodzi $n_{i,1}+n_{i,2} = L$.

Obliczany jest średni rozkład cechy w wylosowanych próbach:

$$p^* = \left(\frac{1}{J}\sum_{i=1}^{J} n_{i,1}, \frac{1}{J}\sum_{i=1}^{J} n_{i,2}\right) = (p^*_1, p^*_2) \tag{5.2}$$

którego wartości podstawiane są następnie do wzoru w celu aproksymowania λ:

$$\lambda \cong \frac{\frac{p^*_1 * Vol(t_k)}{p^*_1 + p^*_2} + card(EEv^+(P,t_k,o))}{\frac{p^*_1 * Vol(t_k)}{p^*_1 + p^*_2} + \frac{p^*_2 * Vol(t_k)}{p^*_1 + p^*_2} + card(EEv^+(P,t_k,o)) + card(EEv^-(P,t_k,o))} \tag{5.3}$$

Jeżeli aproksymowana wartość λ spełnia warunek $0<\lambda\leq\lambda^P$, to zachodzi $KS(t)|=_G Pos(P(o))$, agent generuje komunikat $Pos(P(o))$ jako dobrze ugruntowaną reprezentację wiedzy i kończy pracę.

W przeciwnym przypadku, jeżeli aproksymowana wartość λ spełnia warunek $\lambda^P<\lambda\leq\lambda^B$, to zachodzi $KS(t)|=_G Bel(P(o))$, agent generuje komunikat $Bel(P(o))$ jako dobrze ugruntowaną reprezentację wiedzy i kończy pracę.

W przeciwnym przypadku następuje przejście do **Krok 4**.

Krok 4. Komunikaty wymienione w Tablicy 3.1 nie spełniają relacji dobrego ugruntowania. Algorytm kończy działanie bez wyboru konkretnej reprezentacji lingwistycznej. W tym sensie agent wyposażony w język z tabeli 3.1. milczy.

6. ZAKOŃCZENIE

W pracy podano oryginalny algorytm zachowań językowych kognitywnego agenta. Pokazano w szczególności, w jaki sposób komunikaty generowane przez agenta zależą od wiedzy empirycznej zgromadzonej przez niego w trakcie interakcji z otoczeniem. Zaproponowano sposób modelowania siły przekonań agenta w zakresie stanu pewnej cechy w określonym obiekcie w momencie, gdy nie istnieje możliwość bezpośredniego zweryfikowania jej stanu.

Przedstawiony algorytm jest elementem składowym szerszego modelu zachowań językowych kognitywnych agentów uwzględniającego także inne klasy komunikatów semantycznych. W modelu tym dąży się do zaadaptowania w sztucznych agentach realizowanych

w klasycznym paradygmacie obiektowym jak najszerszej listy założeń dotyczących organizacji modułów językowych. W tym sensie dąży się do połączenia wybranych aspektów podejścia lingwistyki kognitywnej z symbolicznymi metodami sztucznej inteligencji i robotyki.

BIBLIOGRAFIA

[1] Harnad S.: *The symbol grounding problem*, Physica D, 42, pp. 335-346, 1990.

[2] Lakoff G., Johnson M.: *Philosophy in the Flesh: the embodied mind and its challenge to Wester thought*, Basic Books, New York 1999.

[3] Husserl E.: *Logische Untersuchungen, Zweiter Band. Untersuchungen zur Phänomenologie und Theorie der Erkenntnis*, I Teil, Max Niemeyer Verlag, Hlle a.d. Saale 1913.

[4] Husserl E.: *Logische Untersuchungen, Zweiter Band. Elemente einer phänomenologischen Aufklärung der Erkenntnis*, II Teil, Max Niemeyer Verlag, Hlle a.d. Saale 1921.

[5] Katarzyniak R.: *Grounding atom formulas and simple modalities in communicative agents.* W: Proceedings of Applied informatics. The 21st IASTED International Multi-Conference, pp. 388-392.

[6] Katarzyniak R.: *A model for extracting a class of knowledge operators, logic disjunctions and logic alternatives from BDI agents.* W: Advances in Soft Computing Proceedings of the International Symposium Intelligent Information Systems X'2001, Physica-Verlag, pp. 257-269.

[7] Katarzyniak R.: *Intentional models and BDI theories. An inquiry into a relational and logical method of modeling autonomous processes for possible worlds constructions.* W: Intelligent systems & applications. ISA '2000. International ICSC Congress, International Computer Science Conventions, Academic Press, pp. 885-891.

[8] Katarzyniak R.: *Intencjonalny model wiedzy agenta o bieżącym stanie świata.* W: Materiały konferencyjne. Inżynieria wiedzy i systemy ekspertowe 2000, Oficyna Wydawnicza Politechniki Wrocławskiej, str. 142-151.

[9] Katarzyniak R., Pieczyńska-Kuchtiak A.: *Intentional semantics for logic disjunctions, alternatives and cognitive agent's belief.* W: Proceedings of the 14th International Conference on Systems Science, Oficyna Wydawnicza Politechniki Wrocławskiej, pp. 370-382.

[10] Katarzyniak R.: *A conceptual model for extracting implications from data bases in believable agents.* W: Proceedings of the 14th International Conference on Systems Science, Oficyna Wydawnicza Politechniki Wrocławskiej, pp. 382-389.

[11] Katarzyniak R.: *A conceptual model for extracting equivalences from perceptions stored in believable agents* W: 6th Hellenic European Conference on Computer Mathematics and its Applications'2003, Athens, Greece (w druku).

[12] Katarzyniak R., Pieczyńska – Kuchtiak A.: *Strategia wyboru komunikatu modalnego w dialogowym systemie wspomagania decyzji.* Badania Operacyjne i Decyzje, Nr1, 2004 (w druku).

[13] Maruszewski T.: *Psychologia poznawcza.* Polskie Towarzystwo Semiotyczne, Warszawa 1996.

[14] Katarzyniak R., Nguyen N.T.: *Reconciling inconsistent profiles of agent's knowledge states in distributed multiagent systems using consensus methods.* Systems Science, 2000. Vol. 26, No. 4, pp. 93-119.

[15] Katarzyniak R., Pieczyńska-Kuchtiak A.: *A consensus based algorithm for grounding belief formulas in internally stored perceptions.* Neural Network World, 2002, Vol. 5. pp. 461-472.

A MODULE FOR GENERATION OF MODAL ATOM STATEMENTS

Summary

A model for a module responsible for the generation of modal atom statements by a cognitive agent is presented. It is assumed that statements are modal extensions of atom statements of this agent on the state of a property in a particular real world's object. The application of modal operators of knowledge, belief, and possibility is considered. The original model of the agent's mind is accepted in which at least some assumptions of the model of human mind are applied. The so-called grounding relation for modal atom statements is defined. It describes requirements for equivalence of intentional meaning of generated statements and the state of empirical knowledge base stored in the cognitive agent. An example algorithm is given which realizes defined grounding relations. The algorithm refers to simple statistical approximations of the volume of empirical knowledge.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Alina Kmiecik[*], Vincenzo Ambriola[**]

[*]Instytut Informatyki, Politechnika Łódzka
[**]Dipartimento di Informatica, Universitá di Pisa

TRANSFORMACJE MODELU ARCHITEKTONICZNEGO OPROGRAMOWANIA

Streszczenie

Standard OMG Model Driven Architecture (MDA) podniósł znaczenie inżynierii modeli, która z dnia na dzień zdobywa coraz więcej zwolenników. Projektowanie architektoniczne i transformacje stanowią podstawę procesu tworzenia oprogramowania opartego na modelach. Dlatego dla dalszego roz-woju i popularyzacji inżynierii oprogramowania sterowanej modelami niezbędne jest zapewnienie możliwie szerokiego wsparcia dla przekształceń modelowych wykonywanych w ramach typowych zadań architekta. Publikacja przedstawia nasze dotychczasowe prace w zakresie transformacji modeli architektonicznych dedykowanych poprawie parametrów jakościowych i zaspokojeniu wymagań niefunkcjonalnych. Podaje ogólny opis trzech wybranych transformacji i omawia przykład ich zastosowania w celu usprawnienia architektury komercyjnego systemu GIS (*Geographic Information System*).

1. WSTĘP

Analizując procesy produkcyjne w dziedzinie inżynierii lądowej i mechaniki bez trudu można odnotować kluczową rolę projektowania architektonicznego. Wysoki priorytet nadany architekturze produktu wynika z faktu, że niewiele można zmienić w konstrukcji oddanego do eksploatacji budynku lub wyprodukowanej maszyny. Co więcej, każda ewentualna modyfikacja o zasięgu architektonicznym naraża producenta na wysokie koszty oraz ryzyko związane z nieprzewidywalnymi skutkami takiej zmiany (np. ukrytą wadą bądź katastrofą budowlaną). Między innymi z tych względów w budownictwie i mechanice wszelkie zmiany architektoniczne są testowane i przeprowadzane na poziomie modeli a nie na wyjściowych produktach procesu produkcyjnego. Jednakże w Inżynierii Oprogramowania można zaobserwować zgoła odmienne zjawisko. Ogólna fascynacja inżynierią wsteczną (ang. *reverse engineering*) doprowadziła do sytuacji, w której zmiany oprogramowania są definiowane dla kodu, czyli produktu końcowego procesu wytwarzania. Świeżo upieczeni adepci Inżynierii Oprogramowania mogą odnieść wrażenie, iż lepiej jest najpierw postawić budynek a dopiero później przesuwać jego ściany i cegły celem dostosowania jego struktury i parametrów do wymagań użytkowników. W prowadzonych przez nas badaniach z zakresu architektury systemów informatycznych pragniemy wykorzystać doświadczenia

płynące z inżynierii lądowej. Dlatego transformacje architektoniczne definiujemy w ujęciu zmian modeli oprogramowania a nie kodu. Kierując się wskazówkami Kruchtena [6] oraz Jacobsona, Boocha i Rumbaugha [5] dotyczącymi opisu architektury, adaptujemy metamodel UML (*Unified Modeling Language*) i na jego podstawie definiujemy zmiany architektoniczne jako transformacje modeli UML. Do formalnego opisu warunków wstępnych i końcowych transformacji wykorzystujemy język OCL (*Object Constraint Language*) [7]. Mamy świadomość, że UML nie przez wszystkich jest uznawany za język opisu architektury oprogramowania (ang. *Architectural Description Language*, ADL). Niemniej dostrzegamy kilka istotnych cech języka UML, których nie posiada żaden z obecnych języków ADL: jest powszechnie znany i wykorzystywany w środowisku informatycznym do dokumentowania komercyjnych projektów i ma silne wsparcie narzędziowe. Dzięki temu mamy nadzieję wykorzystać doświadczenia praktyków w zakresie projektowania architektonicznego oraz dokonać wiarygodnej analizy wyników działania definiowanych przez nas transformacji.

W publikacji przedstawiamy ogólne założenia dla transformacji modeli architektonicznych i podajemy opis trzech transformacji: podziału klasy, podziału komponentu i przesunięcia komponentu. Ze względu na ograniczenie ilości stron publikacji nie podajemy definicji OCL transformacji odsyłając zainteresowanych do innych opracowań [2]. W zamian szczegółowo omawiamy przykład wykorzystania transformacji do poprawy architektury komercyjnego systemu GIS, który obrazowo wyjaśnia cel i zasadność naszych badań, jak również wykazuje praktyczne zastosowanie transformacji.

2. TRANSFORMACJE ARCHITEKTONICZNE

W [1] wprowadziliśmy trzywarstwową reprezentację architektury, której trzon stanowiły strukturalne elementy opisu systemów informatycznych oprogramowania: węzeł, komponent, interfejs i klasa, uzupełnione opisem związków i zachowania. Zidentyfikowaliśmy podstawowe transformacje architektoniczne i sklasyfikowaliśmy je zależnie od miejsca ich bezpośredniego działania jako:

- *transformacje wysokiego poziomu*, dedykowane topologii węzłów i fizycznej lokalizacji komponentów oprogramowania,
- *transformacje pośredniego poziomu*, wspierające projektowanie wewnętrznej struktury i wzajemnej komunikacji komponentów,
- *transformacje niskiego poziomu*, służące modyfikacji architektury na poziomie klas.

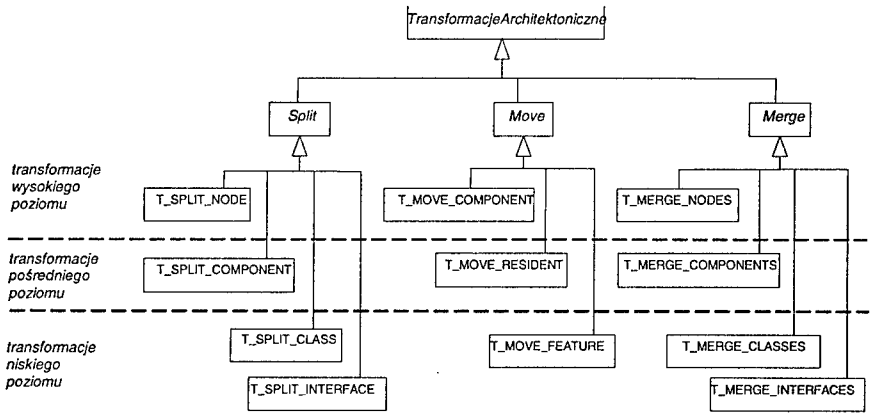
Przyjęliśmy także, że wskazane transformacje stanowią podstawowe narzędzia projektowania architektonicznego a ze względu na pośredni wpływ na charakterystykę oprogramowania (tj. funkcjonalność i wartości atrybutów jakościowych) mogą stanowić koło zamachowe mechanizmów wczesnego zarządzania jakością oprogramowania.

W poniższej publikacji kontynuujemy dotychczasowe podejście przenosząc koncepcję opisu architektonicznego na grunt „4+1" widoków architektonicznych Krutchena (widoki: procesów, fizyczny, logiczny, rozmieszczenia i scenariusze) wpasowanych w ramy metamodelu UML i traktując transformacje architektoniczne jako kompleksowe modyfikacje modelu UML 1.4 [7]. Zakres badań ograniczyliśmy jednak do transformacji zachowujących funkcjonalność modelowanego oprogramowania. Decyzja ta podyktowana została obserwacjami, z których wynika, że najczęściej spotykane wady oprogramowania są skutkiem pominięcia istotnych wymagań niefunkcjonalnych (NFR) i zbyt niskich wartości poszczególnych parametrów jakościowych a nie jak można by przypuszczać – złej funkcjonalności lub niepoprawnych nawyków modelowych zespołu projektowego. Dlatego

za podstawowy cel naszych badań przyjęliśmy wsparcie architekta w transformacjach wstępnie zamodelowanej architektury oprogramowania na rzecz poprawy jakości i zaspokojenia niefunkcjonalnych potrzeb użytkowników. Warunek zachowania zewnętrznej obserwowalnej funkcjonalności wprowadził dodatkowe ograniczenia związane z miejscem aktywacji transformacji. Na poziomie modelu architektonicznego (który z natury procesu projektowego może być niekompletny) nie ma wystarczających przesłanek dla kontroli bezpośrednich zmian zachowania oprogramowania. Dlatego w naszej pracy przyjęliśmy, że transformacje mogą rozpoczynać swoje działanie jedynie z poziomu trzech klasyfikatorów: węzła, komponentu lub klasy. Oczywiście, nie oznacza to, że pozostałe elementy modelu nie są modyfikowane a wewnętrzne zachowanie systemu nie ulega zmianie. Ze względu na pajęczą sieć powiązań elementów modelu pośrednio wiele innych składowych opisu architektury wymaga zmiany podczas działania transformacji (np. powiązania i kolaboracje w których uczestniczy klasyfikator, dowiązane maszyny zmian stanów) w celu zachowania funkcjonalności i spójności modelu UML. Podobnie do definicji proponowanej przez Boscha [3] traktujemy *transformację architektoniczną* jako:

transformację modelu architektury oprogramowania, która prowadzi do nowej wersji modelu architektury oferującej tą samą funkcjonalność.

Zaproponowana definicja przypomina definicję refaktoryzacji [4]. W rzeczywistości każda transformacja modelu architektonicznego jest formą restrukturyzacji dedykowaną pewnej abstrakcji oprogramowania poziomu wyższego niż struktura kodu. Niemniej należy podkreślić, że transformacje modeli architektury nie są metodami refaktoryzacyjnymi diagramów UML [8]. Przede wszystkim różnią się od nich celem, który w przypadku transformacji architektonicznych jest ukierunkowany na zmianę architektury i pośrednio – ogólną poprawę jakości systemu a nie tylko czytelności/przejrzystości kodu. Transformacje architektoniczne mają także inny zasięg oddziaływania. Refaktoryzacja wywodzi się z koncepcji ponownego użycia kodów i ma zastosowanie wobec opisu oprogramowania niskiego poziomu (tj. klas implementacyjnych, ich powiązań i zachowania). Nie dba o zmiany na poziomie topologii węzłów i fizycznej lokalizacji komponentów, co jest niedopuszczalne w przypadku modyfikacji architektonicznych, które – aby mogły być uznane za kompletne – potrzebują objąć zasięgiem wszystkie możliwe widoki architektoniczne. Co więcej, ze względu na ścisłe powiązania elementów modelu architektury transformacje muszą uwzględniać fakt, że zmiana w jednym widoku architektonicznym może wywołać w pozostałych widokach prawdziwą lawinę zmian. Rysunek 1 przedstawia (prawdopodobnie



Rys. 1. Katalog transformacji modeli architektonicznych niezmienniczych funkcjonalnie

jeszcze niepełny) katalog proponowanych transformacji modeli architektonicznych niezmienniczych funkcjonalnie. Elementy zaznaczone kursywą nie stanowią transformacji a jedynie grupy, które gromadzą transformacje o podobnym działaniu lub ograniczeniach.

3. KROK KU FORMALNYM DEFINICJOM TRANSFORMACJI

Model architektury rozpatrujemy w kategoriach języka UML, dlatego naturalną wydaje się decyzja o definiowaniu transformacji jako zmian elementów meta-modelu UML oraz wykorzystanie OCL jako formalnego języka opisu warunków wstępnych i końcowych transformacji. Pragniemy zachować pełną spójność z językiem UML i traktować transformacje jako operacyjne rozszerzenia języka licząc na łatwą integrację narzędzi transformacji z popularnymi narzędziami modelowania opartymi na języku UML. Ze względu na ograniczenia publikacji nie możemy omówić i podać formalnych opisów OCL wszystkich transformacji wyróżnionych na rysunku 1. Poniżej pobieżnie przedstawimy jedynie transformacje użyte w przykładzie (rozdział 4). Zainteresowanych szczegółami definicji transformacji odwołujemy do publikacji [2].

Transformacja podziału klasy T_SPLIT_CLASS(Class: C, set Feature(F)) dzieli klasę C na klasy C i C2 zgodnie z pewną linią podziału wyznaczoną wzdłuż składowych klasy C na podstawie wiedzy o jej zachowaniu np. kolaboracji. (Dobór składowych przenoszonych z klasy C do C2 nie należy do zadań transformacji lecz algorytmu wyższego poziomu realizującego zasadniczy cel zmian architektury). Na pierwszy rzut oka transformacja ma bezpośredni wpływ jedynie na strukturę systemu. Wrażenie to zaciera się, gdy przyjrzymy się bliżej powiązaniom elementów meta-modelu UML. Przykładowo podczas podziału klasy C wszystkie obiekty tej klasy muszą zostać podzielone wzdłuż zbioru instancji atrybutów (ang. slots) przenoszonych do klasy C2; jeśli klasa miała opis zachowania w postaci maszyny zmian stanów, to musi ona zostać przekształcona tak, aby z nowo utworzoną maszyną zmian stanów klasy C2 zapewniała dotychczasową funkcjonalność i spójność modelu; kolaboracje, w których bierze udział klasa C wymagają aktualizacji w kontekście nowo wprowadzanych ról i interakcji. Transformacje niskiego poziomu zazwyczaj wymuszają zmiany w widoku fizycznym architektury. W tym przypadku dla zachowania spójności modelu UML pomiędzy klasą C2 i wszystkimi komponentami, które implementują klasę C muszą zostać utworzone nowe zależności typu <<reside>>.

Transformacja podziału komponentu T_SPLIT_COMPONENT(Component: C, set Residents(R)) jest reprezentantem transformacji poziomu pośredniego i służy do zmiany zakresu implementacyjnego komponentu C. Ze względu na abstrakcyjny charakter modeli UML transformacja ta sprowadza się do podziału zbioru rezydentów (ang. *residents*) klasyfikatora reprezentującego komponent C. Podobnie, jak miało to miejsce w przypadku transformacji podziału klasy, tak i tu szereg innych elementów modelu architektonicznego musi ulec zmianie.

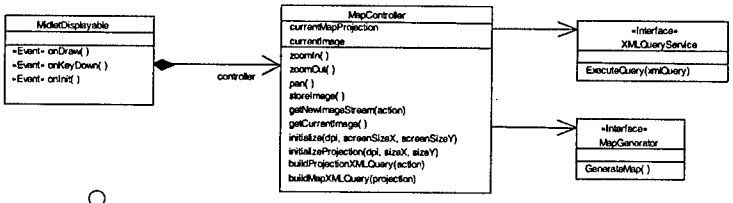
Transformacja przeniesienia komponentu T_MOVE_COMPONENT(Node: N, Node: N2, Component: C) jest stosowana do zmiany lokalizacji danego komponentu w ramach wyznaczonej topologii węzłów. Po przeprowadzeniu transformacji w diagramie rozmieszczenia można zaobserwować zmiany po stronie klientów zależności <<deploy>>, jak również pewne aktualizacje powiązanych z węzłami N i N2 maszyn zmian stanów i kolaboracji.

4. PRZYKŁAD SYSTEMU WEBGIS

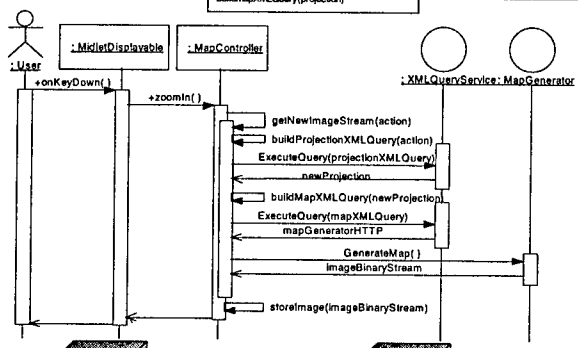
Jednym z powodów, dla których zdecydowaliśmy się użyć języka UML do opisu architektury i zdefiniować zmiany architektoniczne w kontekście zmian modelu była

możliwość testowania praktycznych zastosowań definiowanych transformacji. W naszym eksperymencie wykorzystaliśmy komercyjny system WebGIS[1] oparty na architekturze *klient-serwer* (gruby klient). Zadaniem serwera jest udostępniać serwisy internetowe używane przez aplikacje klienckie do zapewnienia podstawowych funkcji systemu GIS. Klient komunikuje się z serwerem wykorzystując protokół SOAP i wysyłając stosowne zapytanie XML za każdym razem, gdy potrzebuje skorzystać z jakiegoś serwisu. Zapytanie to jest przechwytywane przez komponent *XMLQueryService* i rozsyłane do pozostałych serwisów serwera. W odpowiedzi klient otrzymuje odpowiedź XML wygenerowaną przez właściwy serwis. System *WebGIS* składa się aż z 14 serwisów, jednak ze względu na dopuszczalną ilości stron publikacji poniższą dyskusję ograniczamy do serwisu *MapGenerator*. Zadaniem serwisu *MapGenerator* jest dostarczanie obrazów wybranej mapy kartograficznej zgodnie z parametrami podanymi przez klienta w zapytaniu XML (np. dpi, rozmiar ekranu, typ projekcji kartograficznej). Dla klientów internetowych i stacjonarnych serwis *MapGenerator* uzyskuje bardzo dobą wydajność generując 25 obrazów na sekundę o rozdzielczości 400x400 pikseli.

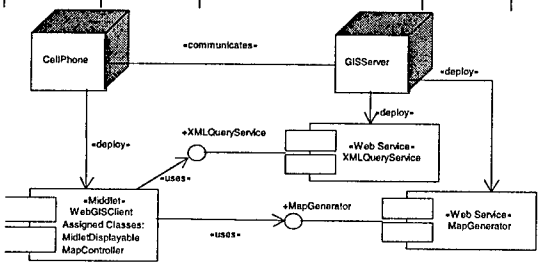
2.a



2.b



2.c



Rys. 2. Uproszczona architektura systemu WebGIS

Firma programistyczna tworząca *WebGIS* postanowiła rozszerzyć zakres działania systemu udostępnić przeglądarkę map kartograficznych również użytkownikom telefonów komórkowych wyposażonych w odbiornik GPRS. Programiści wykorzystali dotychcza-

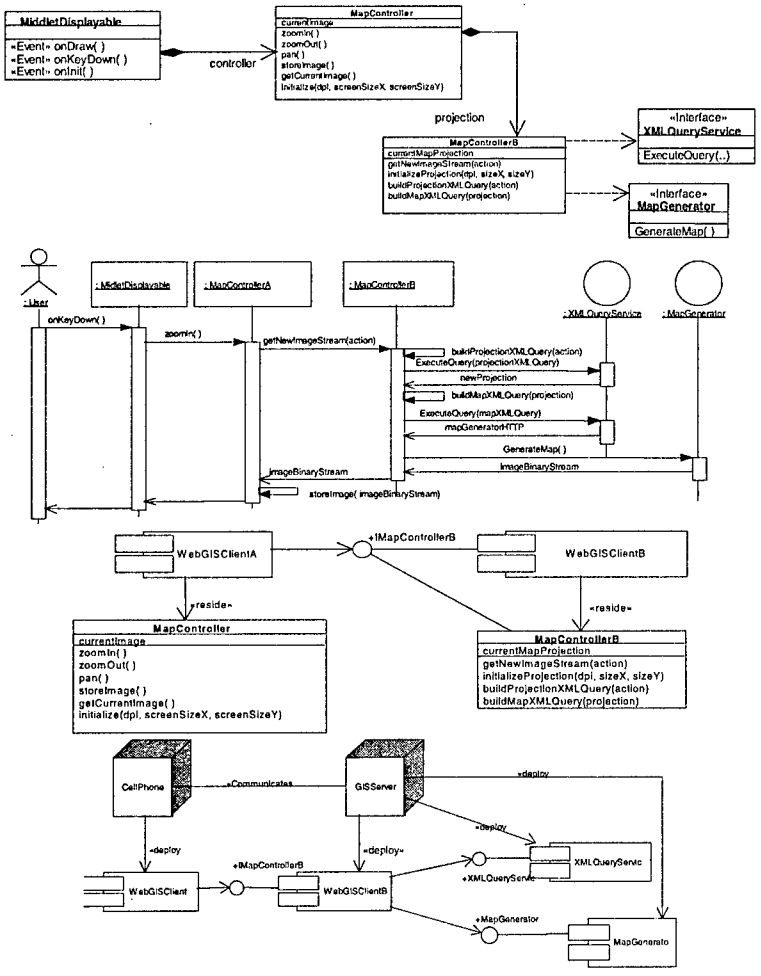
[1] nazwa produktu została zmieniona zgodnie z życzeniem producenta oprogramowania

sową architekturę sprawdzoną w przypadku klientów stacjonarnych i internetowych i zaimplementowali ją bez zmian dla nowej aplikacji klienckiej. Uproszczony model architektury zrekonstruowany na podstawie kodu i uzupełniony wiedzą programistów przedstawiony jest na Rysunku 2 w postaci trzech diagramów UML: 2.a – diagram klas z wyróżnionymi klasami i interfejsami istotnymi z architektonicznego punktu widzenia, 2.b – diagram przebiegu (ang. *sequence diagram*) opisujący istotne interakcje pomiędzy klasami i interfejsami diagramu klas 2.c – diagram rozmieszczenia z wyróżnionymi węzłami klienta i serwera, rozmieszczeniem komponentów i zaimplementowanymi w nich klasami (zależność typu <<reside>>). Jak widać na rysunku, na węźle *CellPhone* działa tylko jeden komponent – middlet *WebGISClient* napisany w języku Java, który implementuje klasy: *MiddletDisplayable*, klasa obsługi interfejsu użytkownika, odpowiedzialna zarówno za wyświetlenie mapy jak i przechwytywanie generowanych przez użytkownika zdarzeń; *MapController*, przechowuje aktualny stan mapy i zapewnia podstawową logikę do nawigacji mapy. Używa do tego celu dwóch interfejsów wystawionych przez serwisy zlokalizowane na serwerze: *XMLQueryService* i *MapGenerator*, które zapewniają operacje właściwe dla projekcji i generowania mapy.

Pomimo, iż powyżej opisana architektura została przetestowana i sprawdzona w przypadku dwóch typów klientów systemu *WebGIS*, była niewłaściwa dla telefonów komórkowych. Okazało się bowiem, że klient musi czekać na pobranie nowej mapy ok. 10 sekund! Blisko miesiąc programiści doszukiwali się przyczyn złej wydajności systemu dopóki nie zdali sobie sprawy, że problem wynika bezpośrednio z zasad funkcjonowania GPRS, który czeka około 3 sekund za każdym razem, gdy inicjalizuje połączenie HTTP i dopiero po tym czasie pozwala na jakikolwiek transfer danych. Z rysunku 2.a i 2.c możemy zauważyć, że komponent *WebGISClient* zlokalizowany w telefonie komórkowym i implementujący klasę *MapController* musi łączyć się aż trzy razy z serwerem (na którym ulokowane są komponenty obsługujące interfejsy *XMLQueryService* i *MapGenerator*) w celu pozyskania mapy. Ponieważ każde połączenie zabiera co najmniej 3 sekundy, sumaryczny czas tłumaczy krytyczne opóźnienie, które spowalnia aplikację klienta.

Stało się oczywiste, że zaimplementowana architektura nie ma zastosowania w przypadku telefonów komórkowych. Ponieważ wada oprogramowania wynikała z liczby komunikatów przepływających pomiędzy klientem i serwerem, to sposobem na poprawę wydajności systemu było istotne zmniejszenie wymiany danych pomiędzy tymi węzłami podczas transferu mapy. Intuicyjnym rozwiązaniem problemu było przesunięcie części odpowiedzialności komponentu *WebGISClient* z węzła *CellPhone* do węzła *WebGISSerwer*. Pierwszy krok w kierunku ulepszonej architektury systemu WebGIS obejmował podział odpowiedzialności klasy *MapController*. W oparciu o diagram przepływu podany na rysunku 2.c wybraliśmy stosowny podzbiór składowych, które powinny zostać przesunięte z klasy *MapController*. Składał się on ze wszystkich tych operacji, które używały interfejsów realizowanych przez komponenty zlokalizowane na serwerze. Mając ustaloną linię podziału klasy *MapController* byliśmy w stanie przeprowadzić transformację T_SPLIT_CLASS. Uzyskaliśmy w jej wyniku dwie klasy *MapController* i *MapControllerB* (rysunek 3.a). Następnie zastosowaliśmy transformację T_SPLIT_COMPONENT wobec komponentu *WebGISClient*, która zaowocowała powstaniem komponentów *WebGISClient* i *WebGISClientB*. Klasy *MiddletDisplayable* i *MapController* pozostały rezydentami (zaimplementowanymi klasami) komponentu *WebGISClient*, natomiast klasa *MapControllerB* została przesunięta do nowo utworzonego komponentu *WebGISClientB*. Należy podkreślić, że komponent *WebGISClientB* musiał udostępnić interfejs *IMapControllerB* realizowany przez klasę *MapControllerB* w celu zapewnienia klasie *MapController* dalsze korzystanie z operacji klasy *MapControllerB*. Wynikowy diagram komponentów zamieszczony jest na rysunku 3.b. Ostatecznie przesunęliśmy komponent *WebGISClientB* z węzła *CellPhone* do

węzła serwera wykorzystując transformację T_MOVE_COMPONENT (rysunek 3c). Dzięki tym transformacjom klient telefonu komórkowego potrzebował połączyć się tylko raz z serwerem w celu otrzymania nowej mapy (rysunek 3.d). Zastosowanie trzech transformacji istotnie poprawiło wydajność systemu: po implementacji zmodyfikowanej architektury klient telefonu komórkowego czekał tylko trzy sekundy na nową mapę.



Rys. 3. Architektura systemu WebGIS po transformacjach

5. PODSUMOWANIE

Opisane w przykładzie „wąskie gardło" oprogramowania GIS w przypadku telefonów komórkowych dotyczyło głównie złej organizacji wewnętrznej komponentów i ich fizycznego rozmieszczenia. Mogło być wykryte dopiero po zestawienia wszystkich trzech poziomów architektonicznego opisu. Mimo, iż krytyczny punkt zachowania systemu był jawnie przedstawiony na diagramie przepływu, nie był uznawany za taki dopóki klasy biorące udział w interakcji nie zostały przypisane implementującym je komponentom.

Zaproponowana modyfikacja dotyczyła trzech widoków architektonicznych: logicznego – ze względu na konieczność podziału odpowiedzialności klasy *MapController*, fizycznego – ze względu na podział klasy *WebGISClient* oraz rozmieszczenia – ze względu na potrzebę zmiany lokalizacji komponentów. Z tych powodów trudno było myśleć o modyfikacji architektonicznej z perspektywy kodu, gdzie brakuje informacji dotyczących rozmieszczenia komponentów. Co więcej, analizując zagadnienie jedynie na poziomie kodu brak było przesłanek, ze organizacja komponentów zimniejsza wydajność systemu. Fakt, że komponenty klienta i serwera zostały napisane w różnych językach programowania (odpowiednio Java i C#) dodatkowo komplikował działanie architekta na poziomie kodu. Przykładowo, proste transformacje kodu nie pozwoliłyby przenieść komponentu Javy z telefonu komórkowego na serwer, gdzie nie ma bibliotek właściwych dla middletów. Z kolei metody refaktoringu przeniesione na poziom modeli UML sprawdziłyby się na poziomie klas, lecz pominęły zmiany architektoniczne wyższego poziomu. Dlatego w przypadku systemu *WebGIS* tylko perspektywa modelu zapewniała właściwy punkt widzenia i elastyczność potrzebną do zidentyfikowania i wykonania niezbędnych poprawek architektonicznych. Nasze obecne prace ukierunkowane są na implementację zintegrowanego narzędzia dla transformacji modeli architektonicznych UML, który usprawni badania ich wpływu na wartości parametrów jakościowych i pozwoli zautomatyzować proces modyfikacji modeli architektonicznych oprogramowania.

BIBLIOGRAFIA

[1] Ambriola V., Kmiecik A.: *Architectural Transformations*, w: Proceedings of the Fourteenth International Conference on Software Engineering and Knowledge Engineering, s. 275-278, ACM 2002.

[2] Ambriola V., Kmiecik A.: *Transformations for Model Restructuring*, Informatica (Słowenia) 2004 (przyjęta do publikacji).

[3] Bosch J., Molin P.: *Software Architecture Design: Evaluation and Transformation*, w: Proceedings of the IEEE Engineering of Computer Based Systems Symposium, IEEE 1999.

[4] Fowler M.: *Refactoring. Improving the Design of Existing Code*, Addison-Wesley, 1999.

[5] Jacobson I., Booch G., Rumbaugh J.: *The Unified Software Development Process*, Addison-Wesley, 1998.

[6] Kruchten F.: *Architectural Blueprints – the "4 +1" View Model of Software Architecture*, s.42-50, IEEE Software 12, 4,1995.

[7] *OMG Unified Modelling Language Specification*, version 1.4, OMG document formal/01-09-67, http://www.uml.org/ , 2001.

[8] Sunye G., Pollet D., Vojtisek D.: *Refactoring UML Models*, w: Proceedings of UML 2001, s. 134-148, LNCS 2185, Springer Verlag.

TRANSFORMATIONS OF SOFTWARE ARCHITECTURE MODELS

Summary

Model-driven engineering reaches more and more followers and gradually grows up as an incoming solution to the software-intensive systems production. Architectural design and model transformations seems to play a fundamental role in this kind of development. This paper presents our efforts in defining model-level architectural transformations. It provides the outline for functionality preserving transformations and gives a closer look to three of them: for component moving, for component splitting, and for class splitting. An example of employing the transformations to remove the architectural bottleneck of industrial WebGIS system is also given.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Marek Kostrzewa

Katedra Informatyki, Akademia Górniczo-Hutnicza

PODEJŚCIE DO JAKOŚCI W METODOLOGIACH WYTWARZANIA OPROGRAMOWANIA

Streszczenie

Tworzenie oprogramowania charakteryzującego się wysoką jakością uzależnione jest od jakości procesu, w trakcie którego powstaje. Metodologia wytwarzania oprogramowania szczegółowo opisuje proces jego powstawania. Jednakże nie istnieje jednostka miary jakości dająca się wyrazić pojedynczą wartością liczbową. W niniejszym referacie wykorzystano model jakości oprogramowania ISO 9126 i poddano analizie kluczowe praktyki wybranych metodologii wytwarzania oprogramowania z punktu widzenia realizacji tego modelu.

1. WSTĘP

Tworzenie złożonych systemów informatycznych to proces obarczony wysokim stopniem ryzyka niepowodzenia. Przyczyna tego ryzyka leży w samej naturze tworzonego systemu, gdyż często końcowy produkt nie jest dostępny do bezpośredniej percepcji, nierzadko jest unikatowy, a ponadto charakteryzuje się znacznym rozmiarem oraz wysoką złożonością struktury i dynamiki.

Firmy wytwarzające oprogramowanie, które chcą być konkurencyjne na rynku, zmuszone są do podnoszenia jakości tworzonych systemów. Zapewnienie jakości tworzonego systemu opiera się na założeniu, że jakość produktu jest silnie związana z jakością procesu, w trakcie którego powstaje produkt. W przypadku wytwarzania oprogramowania, przez jakość rozumie się miedzy innymi optymalizację zarządzania produkcją, ścisłą realizację projektu zgodnie z harmonogramem oraz wymaganiami klienta oraz wypracowanie strategii wdrażania i szeroko rozumianej konserwacji oprogramowania. Ryzyko związane z niepowodzeniem, które jest obecne podczas tworzenia produktu ma istotny wpływ na jego końcową jakość. Często stara się zmniejszyć ryzyko kosztem akceptacji gorszej jakości końcowego systemu. Pomimo unikalności tworzonych systemów informatycznych opracowano wiele metodologii wspomagających ich wytwarzanie. Kluczowym etapem każdego przedsięwzięcia informatycznego jest wybór odpowiedniej metodologii z ogółu dostępnych. Od wybranej metodologii oczekuje się, że będzie przystosowana do realiów tworzonego systemu, ale przede wszystkim powinna ona posiadać dopracowane podejście do szeroko rozumianej jakości wytwarzania.

Celem referatu jest charakterystyka wybranych metodyk pod kątem możliwości tworzenia oprogramowania charakteryzującego się wysoką jakością.

2. JAKOŚĆ OPROGRAMOWANIA

Inżynieria oprogramowania nie znalazła jednoznacznej odpowiedzi na pytanie czym jest jakość oprogramowania. Nie istnieje i nie wydaje się, by była kiedykolwiek możliwa do wprowadzenia jednostka miary jakości dająca się wyrazić pojedynczą wartością liczbową. „Jakość jest trudna do zdefiniowania i niemożliwa do zmierzenia aczkolwiek łatwa do rozpoznania" [1]. Gilles [2] ponadto zauważa, że: „Jakość jest zwykle niewidoczna kiedy jest obecna, aczkolwiek łatwo zauważalna jest jej nieobecność". Próby podania definicji pojęcia jakości można znaleźć też w [3, 4, 5, 6]. Zgodnie z [5] w inżynierii oprogramowania istnieją dwie interpretacje pojęcia jakości, a mianowicie:

1. stopień zgodności z podanymi wymaganiami,
2. stopień zgodności z potrzebami i oczekiwaniami klienta bądź użytkownika.

Pierwsza z powyższych interpretacji zakłada, iż jakość jest zdefiniowana pośrednio bądź bezpośrednio w sprecyzowanych wymaganiach. Natomiast druga interpretacja nie zakłada obecności formalnie podanej definicji jakości produktu, lecz pozostawia tą kwestię do subiektywnego rozstrzygnięcia przez klienta czy użytkownika (tym samym ukazuje relatywizm związany z jakością – jakość jest względna w stosunku do konkretnych oczekiwań).

W tej pracy jakość oprogramowania przyjmuje się zgodnie z modelem zaprezentowanym w standardzie ISO 9126 [7], w którym jakość została określona poprzez zbiór 27 atrybutów pogrupowanych w 6 kategoriach [tablica 2.1].

Dopełnieniem modelu jakości oprogramowania stanowią metryki wykorzystywane do pomiaru jakości oprogramowania zgodnie ze standardem ISO 9126. Pogrupowane zostały na trzy grupy:

- metryki zewnętrzne [8],
- metryki wewnętrzne [9],
- metryki jakości użycia oprogramowania [10].

Tablica 2.1

Model jakości oprogramowania ISO 9126

Kategoria	*Atrybuty*
Funkcjonalność (ang. Functionality)	Odpowiedniość (ang. suitability)
	Poprawność (ang. accuracy)
	Niezależność (ang. interoperability)
	Bezpieczeństwo (ang. security)
	Zgodność funkcjonalna (ang. functionality compliance)
Niezawodność (ang. Reliability)	Dojrzałość (ang. maturity)
	Odporność (ang. fault tolerance)
	Zdolność powrotu do poprzedniego stanu (ang. recoverability)
	Standard niezawodności (ang. reliability compliance)

Użyteczność (ang. Usability)	Zrozumiałość (ang. understandability)
	Przystępność (ang. learnability)
	Łatwość obsługi (ang. operability)
	Atrakcyjność (ang. attractiveness)
	Standard użyteczności (ang. usability compliance)
Wydajność (ang. Efficiency)	Ograniczoność czasowa (ang. time behavior)
	Zarządzanie zasobami (ang. resource utilisation)
	Standard wydajności (ang. efficiency compliance)
Pielęgnacja (ang. Maintainability)	Zdolność analizy (ang. analyzability)
	Zdolność modyfikacji (ang. changeability)
	Stabilność (ang. stability)
	Zdolność weryfikacji (ang. testability)
	Standard pielęgnacji (ang. maintainability compliance)
Zdolność przenoszenia (ang. Portability)	Zdolność adaptacji (ang. adaptability)
	Możliwość instalacji (ang. installability)
	Współistnienie (ang. co-existence)
	Zastępowalność (ang. replaceability)
	Standard przenoszenia (ang. portability compliance)

2.1. Funkcjonalność

Funkcjonalność to kategoria charakteryzująca oprogramowanie pod kątem zgodności z wymaganiami funkcjonalnymi. Określona jest przez następujące atrybuty:

odpowiedniość – określa obecność funkcji oferowanych przez oprogramowanie, które realizują zadania oczekiwane od tego oprogramowania,

poprawność – określa zdolność oprogramowania do otrzymywania oczekiwanych rezultatów,

niezależność – określa zdolność oprogramowania do realizowania swoich funkcji niezależnie od otoczenia (tj. inne oprogramowanie), w którym został zainstalowany,

bezpieczeństwo – określa zdolność oprogramowania do ochrony wykorzystywanych danych przed nieautoryzowanym dostępem,

zgodność funkcjonalna – określa stopień zgodności oprogramowania z obowiązującymi standardami związanymi z funkcjonalnością.

2.2. Niezawodność

Niezawodność jest kategorią charakteryzującą oprogramowanie pod kątem możliwości utrzymania pewnego poziomu wydajności przy założonych warunkach. Określona jest przez następujące atrybuty:

dojrzałość – określa zdolność oprogramowania do realizowania swoich funkcji, pomimo ewentualnych błędów występujących w tym oprogramowaniu,

odporność – określa zdolność oprogramowania do utrzymania określonego poziomu wydajności oraz funkcjonalności pomimo wystąpienia błędu,

zdolność powrotu do poprzedniego stanu – określa zdolność oprogramowania do odtworzenia określonego poziomu wydajności bądź też odzyskania utraconych danych z powodu wystąpienia błędu,

standard niezawodności – określa stopień zgodności oprogramowania z obowiązującymi standardami związanymi z niezawodnością.

2.3. Użyteczność

Użyteczność to kategoria, która charakteryzuje oprogramowanie pod kątem zrozumiałości, przystępności oraz atrakcyjności. Określona jest przez następujące atrybuty:

zrozumiałość – określa łatwość zrozumienia przez użytkownika przeznaczenia oprogramowania oraz sposobu jego użycia do poszczególnych zadań,

przystępność – określa łatwość nauczenia się przez użytkownika obsługi oprogramowania,

łatwość – określa możliwość obsługi oraz kontroli oprogramowania przez użytkownika,

atrakcyjność – określa atrakcyjność oprogramowania dla użytkownika,

standard użyteczności – określa stopień zgodności oprogramowania z obowiązującymi standardami związanymi z użytecznością.

2.4. Wydajność

Wydajność to kategoria charakteryzująca oprogramowanie pod kątem otrzymywanych osiągów w stosunku do wykorzystywanych zasobów. Określona jest przez następujące atrybuty:

ograniczoność czasowa – określa zdolność oprogramowania do wykonywania swoich funkcji w założonym czasie przetwarzania lub odpowiedzi,

zarządzanie zasobami – określa zdolność oprogramowania do wykorzystywania określonych zasobów podczas wykonywania swoich funkcji,

standard wydajności – określa stopień zgodności oprogramowania z obowiązującymi standardami związanymi z wydajnością.

2.5. Pielęgnacja

Kategoria ta charakteryzuje oprogramowanie pod kątem możliwości przeprowadzania modyfikacji tj. naprawy błędów, dokonywania ulepszeń czy też adaptacji w oprogramowaniu. Określona jest przez następujące atrybuty:

zdolność analizy – określa możliwość przeprowadzenia diagnostyki oprogramowania w celu wykrycia przyczyny błędu, bądź też zidentyfikowania części oprogramowania, która zostanie poddana modyfikacji,

zdolność modyfikacji – określa możliwość wykonania określonych modyfikacji,

stabilność - określa zdolność oprogramowania do zachowania funkcjonalności pomimo przeprowadzenia określonej modyfikacji.

zdolność weryfikacji – określa zdolność oprogramowania do przeprowadzenia weryfikacji dokonanych modyfikacji,

standard pielęgnacji – określa stopień zgodności oprogramowania z obowiązującymi standardami związanymi z pielęgnacją.

2.6. Zdolność przenoszenia

Kategoria ta charakteryzuje oprogramowanie pod kątem możliwości przenoszenia z danego środowiska (np. system operacyjny) do innego. Określona jest przez następujące atrybuty:

zdolność adaptacji – określa zdolność oprogramowania do adaptacji do określonego środowiska,

możliwość instalacji – określa zdolność oprogramowania do instalacji w określonym środowisku,

współistnienie – określa zdolność oprogramowania do współistnienia w jednym środowisku z innymi niezależnymi oprogramowaniami dzieląc wspólne zasoby,

zastępowalność – określa zdolność oprogramowania do zastąpienia go innym w danym środowisku,

standard przenoszenia – określa stopień zgodności oprogramowania z obowiązującymi standardami związanymi ze zdolnością do przenoszenia.

3. METODOLOGIE WYTWARZANIA OPROGRAMOWANIA

Celem metodologii jest podanie metody wytwarzania oprogramowania. Charakterystykę wybranych metodologii można znaleźć w [11, 12, 13, 14, 15, 16]. Przyjęto podział metodologii na tzw. ciężkie oraz lekkie. Za przykład metodologii ciężkiej wybrano Rational Unified Process (RUP), natomiast za przykład lekkiej – Extreme Programming (XP).

3.1. Rational Unified Process (RUP)

Metodyka oparta na iteracyjno-przyrostowym procesie wytwarzania oprogramowania. Szczegółowy opis tej metodyki znajduje się w [15]. Korzysta z sześciu kluczowych praktyk tworzenia oprogramowania:

- iteracyjne wytwarzanie oprogramowania,
- zarządzanie wymaganiami,
- wykorzystywanie architektury komponentowej,
- modelowanie wizualne,
- weryfikacja jakości oprogramowania,
- kontrola modyfikacji w oprogramowaniu.

3.2. Extreme Programming (XP)

Metodyka, w której nacisk położono na zdolność adaptacji wykorzystywanych procesów do możliwych zmian oraz na ścisłą współpracę członków zespołu tworzącego oprogramowanie pomiędzy sobą oraz końcowym użytkownikiem[16]. Wykorzystuje 12 kluczowych praktyk tworzenia oprogramowania:

- zaangażowanie klienta,
- iteracyjne wytwarzanie oprogramowania,
- częste wydawanie wersji oprogramowania,
- prostota architektury,
- intensywne testowanie
- refaktoryzacja tworzonego kodu źródłowego,
- programowanie parami,

- współdzielenie i współodpowiedzialność za stworzony kod źródłowy,
- częsta integracja wytworzonego kodu źródłowego,
- stałe tempo wytwarzania oprogramowania,
- stosowanie standardów tworzonego kodu źródłowego.
- metafora systemu.

4. JAKOŚĆ W METODOLOGIACH WYTWARZANIA OPROGRAMOWANIA

Poniżej przedstawiono rezultat analizy kluczowych praktyk tworzenia oprogramowania pochodzących z RUP [tabela 4.1] oraz XP [tabela 4.2] pod kątem realizacji jakości zgodnie z modelem ISO 9126.

Tablica 4.1

Realizacja jakości w RUP

Iteracyjne wytwarzanie oprogramowania	Funkcjonalność (odpowiedniość, poprawność, bezpieczeństwo)
	Niezawodność (dojrzałość, odporność)
	Wydajność (ograniczoność czasowa, zarządzanie zasobami)
Zarządzanie wymaganiami	Funkcjonalność (odpowiedniość)
	Użyteczność (zrozumiałość)
	Pielęgnacja (zdolność analizy)
Modelowanie wizualne	Funkcjonalność (odpowiedniość)
	Użyteczność (zrozumiałość, przystępność)
	Pielęgnacja (zdolność analizy, standard pielęgnacji)
Wykorzystywanie architektury komponentowej	Funkcjonalność (niezależność)
	Pielęgnacja (zdolność analizy, modyfikacji, weryfikacji, standard pielęgnacji)
	Zdolność przenoszenia (zdolność adaptacji, możliwość instalacji, współistnienie, zastępowalność, standard przenoszenia)
	Użyteczność (zrozumiałość, przystępność)
	Pielęgnacja (zdolność analizy)
Weryfikacja jakości oprogramowania	Funkcjonalność (odpowiedniość, poprawność, bezpieczeństwo)
	Niezawodność (dojrzałość, odporność)
	Wydajność (ograniczoność czasowa, zarządzanie zasobami)
	Pielęgnacja (zdolność weryfikacji, stabilność)
Kontrola modyfikacji w oprogramowaniu	Funkcjonalność (odpowiedniość, poprawność)
	Pielęgnacja (zdolność modyfikacji oraz weryfikacji, stabilność)

Tablica 4.2

Realizacja jakości w XP

Zaangażowanie klienta	Funkcjonalność(odpowiedniość, poprawność)
	Użyteczność (zrozumiałość, przystępność, łatwość obsługi, atrakcyjność)
Iteracyjne wytwarzanie oprogramowania	Funkcjonalność (odpowiedniość, poprawność, bezpieczeństwo)
	Niezawodność (dojrzałość, odporność)
	Wydajność (ograniczoność czasowa, zarządzanie zasobami)
Częste wydawanie wersji oprogramowania	Funkcjonalność (odpowiedniość, poprawność, bezpieczeństwo)
	Niezawodność (dojrzałość, odporność)
	Wydajność (ograniczoność czasowa, zarządzanie zasobami)
	Użyteczność (zrozumiałość, przystępność, łatwość obsługi, atrakcyjność)
Prostota architektury	Pielęgnacja (zdolność analizy, zdolność modyfikacji, zdolność weryfikacji)
Intensywne testowanie	Funkcjonalność (odpowiedniość, poprawność, bezpieczeństwo)
	Niezawodność (dojrzałość, odporność)
	Wydajność (ograniczoność czasowa, zarządzanie zasobami)
	Pielęgnacja (zdolność weryfikacji, stabilność)
Refaktoryzacja tworzonego kodu źródłowego	Pielęgnacja (zdolność modyfikacji)
Programowanie parami	Pielęgnacja (zdolność modyfikacji)
Współdzielenie i współodpowiedzialność za stworzony kod źródłowy	Pielęgnacja (zdolność modyfikacji)
Częsta integracja wytworzonego kodu źródłowego	Pielęgnacja (stabilność)
Stałe tempo wytwarzania oprogramowania	–
Stosowanie standardów tworzonego kodu źródłowego	Pielęgnacja (zdolność analizy, zdolność modyfikacji, standard pielęgnacji)
Metafora systemu	Użyteczność (zrozumiałość, przystępność)

5. ZAKOŃCZENIE

W rozważanych metodologiach dostrzegalny jest podobny dość wysoki nacisk na jakość oprogramowania. Zauważalna jest także nieznaczna przewaga RUP nad XP w pełniejszym realizowaniu standardu modelu jakości ISO 9126. Wynika to pośrednio z przyjętych założeń XP, w którym nacisk położono na intensywną współpracę z końcowym użytkownikiem oraz prostotę wykorzystanych rozwiązań, nadając czystości tworzonej architektury drugorzędne znaczenie. Natomiast RUP, w którym dobrze dobrana i przemy-

ślana architektura pełni istotną rolę, wykorzystuje zalety płynące z wykorzystania architektury komponentowej np. możliwość realizacji miedzy innymi zdolności przenoszenia oprogramowania.

BIBLIOGRAFIA

[1] Kitchenham B.: *Software Metrics*. In Software Reliability Handbook, Elsevier, 1989.
[2] Gillies, Alan C.: *Software Quality, Theory and Management*, International Thomson Computer Press, 1997.
[3] Schulmeyer, G. Gordon and McManus, James I.: *Handbook of Software Quality Assurance*, 3rd Edition, Prentice Hall PRT, 1998.
[4] Petrasch, R.: *The Definition of 'Software Quality': A Practical Approach*. In: Proceedings of the 10th International Symposium on Software Reliability Engineering, Fast Abstracts & Industrial Practices, Nov. 1999, IEEE Computer Society, 1999.
[5] IEEE Standards Collection, Software Engineering.: *IEEE Standard Glossary of Software Engineering Terminology, Std 610-12-1990*. IEEE Computer Society, 1994.
[6] Wallmuller E.: *Software Quality Assurance - A Practical Approach*, Prentice Hall, 1994.
[7] International Standard ISO/IEC 9126-1: *Software engineering-Product quality-Part 1: Quality model*, International Organization for Standardization/International Electrotechnical Commission. Geneva, 2001.
[8] International Standard ISO/IEC 9126-2: *Software engineering–Product quality–Part 2: External metric*, International Organization for Standardization/International Electrotechnical Commission. Geneva, 2001.
[9] International Standard ISO/IEC 9126-3: *Software engineering–Product quality–Part 3: Internal metrics*, International Organization for Standardization/International Electrotechnical Commission. Geneva, 2001.
[10] International Standard ISO/IEC 9126-4: *Software Engineering - Software product quality – Part 4: Quality in use metrics*, International Organization for Standardization/International Electrotechnical Commission. Geneva, 2001.
[11] Boehm B. W.: *Software Engineering*, IEEE Transactions on Computers **C-25**,1976.
[12] Sommerville I.: *Software Engineering*, Addison-Wesley, 1992.
[13] Szejko S. (redakcja): *Metody wytwarzania oprogramowania*, MIKOM, 2002.
[14] Jaszkiewicz A.: *Inżynieria oprogramowania*, Helion, 1997.
[15] Kruchten P.: *The Rational Unified Process, An Introduction, Second Edition*, Addison Wesley Longman, 2000
[16] Beck K.: *Extreme Programming*, Addison Wesley Longman, 2000

QUALITY IN METHODOLOGY OF SOFTWARE DEVELOPMENT

Summary

High quality software development depends on quality of production's process. Software methodology specifies in detail creation of software. There is no unit of quality measure that can be express by a single value. Quality model ISO 9126 was used to consider key practices of selected software methodologies related to software quality.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Jarosław Kuchta

Katedra Architektury Systemów Komputerowych, Politechnika Gdańska

INTEGRACJA MODELOWANIA I IMPLEMENTACJI OPROGRAMOWANIA W METODZIE IMAC

Streszczenie

Dotychczasowa technologia wytwarzania oprogramowania akceptuje występowanie niespójności pomiędzy modelowaniem w języku UML a implementacją w obiektowych językach programowania. Metoda IMAC zapewnia spójność przez zastosowanie jednolitej notacji graficznej w całym projekcie informatycznym. Dzięki temu implementacja zostaje sprowadzona do uszczegółowienia diagramów projektowych i wygenerowania gotowego, funkcjonalnego kodu programu. Niniejsza praca prezentuje skrócone zasady notacji IMAC oraz przykład wykorzystania diagramów interakcji w czasie implementacji w kontekście metody IMAC.

1. WPROWADZENIE

IMAC (ang. *Integrated Method of Application Construction*) jest metodą wytwarzania oprogramowania umożliwiającą spójne połączenie modelowania analitycznego i projektowego z implementacją. Stosowane do tej pory technologie informatyczne akceptowały występowanie pewnej niespójności pomiędzy modelowaniem a implementacją. Model wykonywany w graficznym języku UML [1] przedstawia najpierw wymagania systemowe, a później koncepcje projektowe w postaci diagramów. Z kolei implementacja wymaga napisania kodu programu w wybranym języku programowania. Niespójność przejścia od modelowania do implementacji polega na zmianie reprezentacji projektu z graficznej na tekstową, zmianie aparatu pojęciowego, zmianie narzędzi pracy.

Metoda IMAC usuwa niespójności między modelowaniem a implementacją przez zastosowanie jednolitej, graficznej notacji całego projektu informatycznego. IMAC zapewnia też właściwe odwzorowanie pomiędzy aparatem pojęciowym stosowanym podczas modelowania a aparatem pojęciowym z dziedziny implementacji.

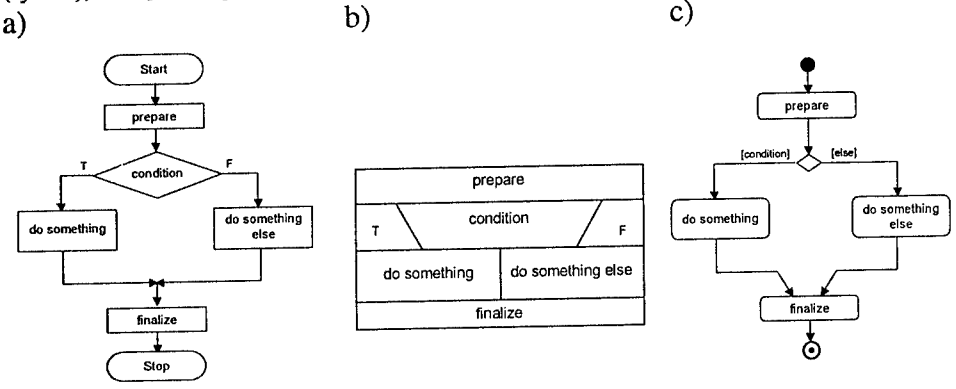
W niniejszej pracy najpierw poznamy skrócone zasady jednolitej notacji IMAC a następnie wykorzystanie diagramów funkcjonalnych UML do implementacji w kontekście metody IMAC.

2. SKRÓCONE ZASADY NOTACJI IMAC

Metoda IMAC zakłada jak największą zgodność z dotychczas wykorzystywanymi metodami: z UML[2] i ze współczesnymi językami obiektowymi [3][4][5]. UML jednak

był tworzony pod kątem modelowania, a nie implementacji. Istnieją w nim co prawda tzw. diagramy implementacji, ale opisują one jedynie pewne aspekty implementacji, jak struktura komponentów i przypisanie komponentów do węzłów przetwarzania w czasie wykonania. Poziom szczegółowości możliwy do wyrażenia w tych diagramach jest właściwy dla fazy projektowania. Istnieją inne diagramy, które mogłyby zostać wykorzystane podczas implementacji, jak np. diagramy stanów, diagramy aktywności, diagramy interakcji. Brak jest w nich jednak wielu elementów właściwych dla tej fazy, jak np. proces, iteracja, skok, przypisanie wartości. Metoda IMAC uzupełnia te szczegóły, a dodatkowo zapewnia stosowanie jednolitej notacji w całym procesie wytwarzania oprogramowania.

W metodzie IMAC jako źródło notacji wykorzystano wykresy operacyjne (rys.1a). Wykres operacyjny ma tę podstawową zaletę, że jest prosty i czytelny. Poza tym jest na tyle uniwersalny, że pozwala na przedstawienie dowolnego algorytmu [6]. Jego wadą jest jednak bardzo duża swoboda projektowania, która umożliwia łatwe naruszanie zasad programowania strukturalnego wykorzystywanych do dziś we współczesnych językach programowania. W latach 70-tych w celu przedstawienia algorytmów strukturalnych opracowano notację strukturalną zwaną diagramami Nassi-Shneidermanna w skrócie N-S [7] lub Chapina [8] (rys.1b). Diagramy te nie znalazły szerokiego uznania ze względu na swoją małą elastyczność i małą czytelność. Jednak mają bardzo dużą zaletę: struktura diagramu odpowiada strukturze kodu. W metodzie IMAC połączono elastyczność wykresów operacyjnych z możliwościami tworzenia algorytmów strukturalnych. Wykorzystano te elementy wykresów operacyjnych, które można odnaleźć diagramach aktywności UML (rys.1c), a więc stan początkowy i końcowy, proces (aktywność) i warunek.

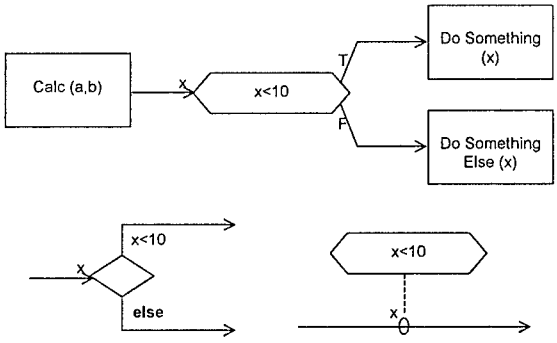


Rys. 1. Źródła notacji metody IMAC: a) wykres operacyjny, b) diagram N-S, c) diagram aktywności UML

Proces w metodzie IMAC jest jednostką funkcjonalności, reprezentuje zbiór operacji wykonywanych w niepodzielnym (w danej skali) czasie i jest w IMAC wyrażany przez prostokąt. Warunek jest zdefiniowany jako osobny element modelowy reprezentujący stereotypowane wyrażenie logiczne. Stereotypami warunku mogą być: **if**, **while**, **until**, **repeat**, **for**, **switch**.

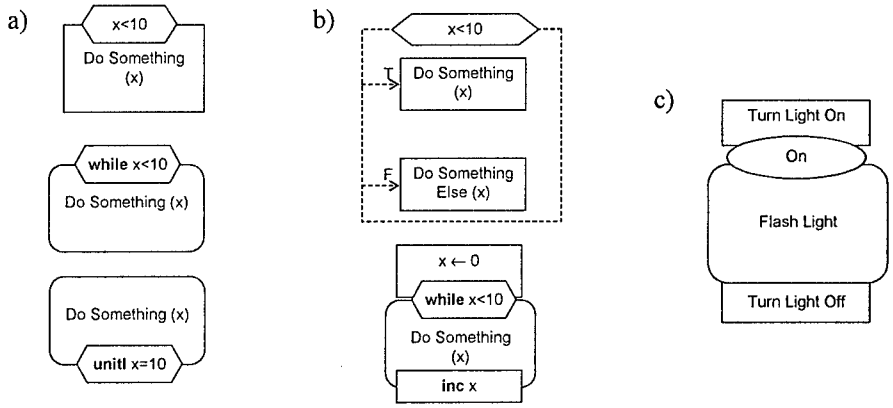
Warunek jest wyrażany przez wydłużony sześciokąt (⬭). Taka forma została przyjęta, aby ułatwić wpisywanie wyrażenia do wnętrza warunku. Jeśli wewnątrz warunku nie będzie umieszczone żadne wyrażenie, to może on zostać zredukowany do formy rombu (◇). Warunek może zostać umieszczony na drodze przepływu sterowania w diagramach aktywności, na drodze przejścia stanów w diagramach stanów, na drodze przepływu

komunikatów w diagramach interakcji. Ponieważ w ciągu przepływu może zostać umieszczonych wiele warunków (tworzących rozgałęzienia), uzyskujemy tym samym większą elastyczność projektowania niż przy pomocy samego warunku strzegącego w UML. Jeśli w narysowanym już diagramie brak jest miejsca na umieszczenie warunku na drodze przepływu, to można (o ile warunek nie powoduje rozgałęzienia) dołączyć warunek do linii przepływu przez tzw. konektor – linię przerywaną zakończoną „oczkiem" (rys. 2).



Rys.2. Przykłady użycia warunku

Warunek może też zostać umieszczony na górnej krawędzi ramki procesu. Wraz z procesem tworzy on wówczas tzw. *strukturę sterowania*. Jeśli warunek będzie stereotypu domyślnego **if**, to proces zostanie wykonany wówczas, gdy warunek zostanie spełniony. Jeśli warunek będzie stereotypu **while**, **until**, **repeat** lub **for**, to wykonanie procesu będzie powtarzane zgodnie ze stereotypem i treścią warunku. Proces taki powinien przyjąć formę aktywności – prostokąta z zaokrąglonymi rogami, bowiem wówczas wykonanie procesu trwa przez pewien czas i może zostać przerwane na skutek warunków wewnętrznych. Warunek stereotypu **while** lub **until** można umieścić również na dolnej krawędzi ramki aktywności. Oznacza to wówczas, że sprawdzanie warunku następuje *po* każdorazowym iteracyjnym wykonaniu procesu.



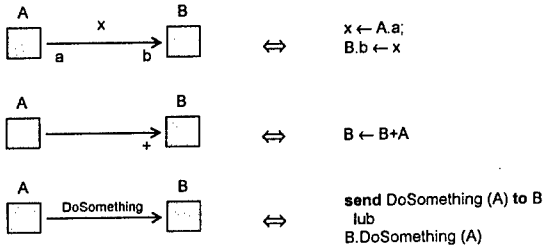
Rys. 3. Przykłady struktur sterowania: a) proste struktury sterowania, b) złożone struktury sterowania, c) wykorzystanie struktur sterowania na diagramach stanów

Wyrażenie tym sposobem instrukcji **if-then-else** wymaga trochę bardziej złożonej struktury. Ponieważ umieszczenie warunku na krawędzi ramki procesu uzależnia cały ten proces, więc konieczne jest wprowadzenie jeszcze jednego elementu modelowego – ramki grupującej. Będzie ona reprezentowała grupę elementów powiązanych semantycznie, lecz nie powiązanych czasem wykonania. Będzie ona wyrażana graficznie przez prostokąt przerywany. Wewnątrz ramki grupującej umieszcza się ramki procesów. Od krawędzi ramki do wewnętrznych ramek procesów prowadzi się odpowiednio oznaczone przerywane strzałki.

Struktury sterowania mogą być jeszcze bardziej złożone. Umieszczenie ramki procesu pod górną krawędzią ramki warunku oznacza akcję wykonywaną przed sprawdzeniem warunku. Umieszczenie ramki procesu pod górną lub pod dolną krawędzią ramki aktywności oznacza akcję wykonywaną odpowiednio przed rozpoczęciem aktywności lub zakończeniu aktywności. Odpowiednie umieszczenie ramki procesu na górnej lub na dolnej krawędzi ramki aktywności oznacza akcję wykonywaną przy każdej iteracji.

W metodzie IMAC struktury sterowania mogą występować nie tylko na diagramach przepływu sterowania, ale też na diagramach stanów. Dla odróżnienia stanu od aktywności potrzebny stał się inny symbol na oznaczenie stanu. W UML stan oznacza się tak samo jak aktywność – przez prostokąt zaokrąglony. W IMAC odróżnia się stan od aktywności oznaczając ten pierwszy symbolem elipsy. Jeśli dla danego stanu jest zdefiniowana aktywność, to elipsę stanu umieszcza się na górnej krawędzi ramki aktywności. Akcje wejściowe i wyjściowe umieszcza się odpowiednio pod górną i dolną krawędzią elipsy stanu (ramki aktywności).

Kolejnym elementem zdefiniowanym w IMAC jest ikonid. Jest to przedstawienie obiektu lub klasy składające się z nazwy obiektu lub klasy oraz ikony. W IMAC ikony używane są odważniej niż w UML. Każda klasa, nie tylko stereotyp, może mieć własną ikonę. Ikony podlegają dziedziczeniu, podobnie jak inne cechy klasy. Obiekt jest przedstawiany przez ikonę własnej klasy i własną nazwę (rys. 4).



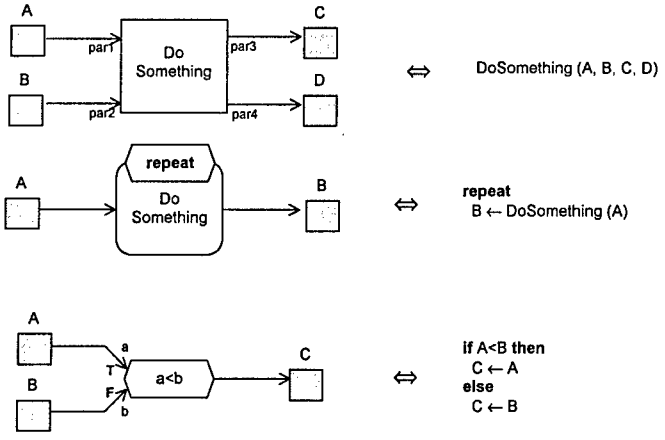
Rys. 4. Przepływ danych między obiektami

W metodzie IMAC ujednolicono wektory przepływu. Przypomnijmy, w UML stosuje się przejścia stanów i przepływy komunikatów. Na diagramach implementacyjnych potrzebny jest przepływ sterowania i przepływ danych. Nie ma jednak konieczności wprowadzania czterech odrębnych rodzajów wektorów przepływu. Przejście stanów można potraktować jak przejście sterowania występujące w kontekście stanów. Podobnie przepływ komunikatów można uznać za specjalny przypadek przepływu danych, który reprezentuje przepływ dowolnej informacji.

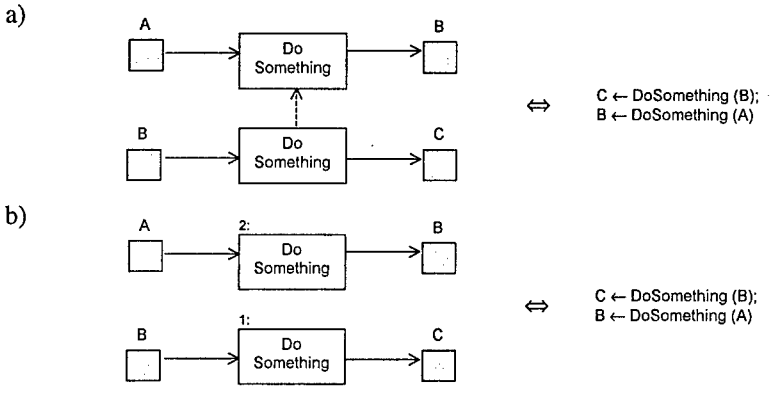
Stosując zasadę jednolitej notacji trzeba zapewnić, by reprezentacja przepływu sterowania i przepływu danych na wszystkich diagramach była taka sama oraz by przepływ sterowania był odróżnialny od przepływu danych. Dlatego w metodzie IMAC przyjęto, że

przepływ danych będzie oznaczany strzałką ciągłą, a przepływ sterowania strzałką przerywaną. Przepływ danych może łączyć symbole obiektów, ramki procesów, warunki, aktywności.

W UML wektor przepływu komunikatu przesyłany od obiektu do obiektu jest opisywany sygnaturą operacji obiektu docelowego i oznacza żądanie wykonania operacji. W IMAC wektor taki może mieć trzy opisy: na końcu (przy obiekcie docelowym), w środku wektora oraz na początku (przy obiekcie źródłowym). Opis końcowy może być sygnaturą operacji lub nazwą atrybutu obiektu docelowego, w tym drugim przypadku oznacza żądanie zapisania przesyłanej informacji do atrybutu. Opis środkowy oznacza przesyłaną informację i może być nazwą komunikatu lub nazwą zmiennej lokalnej. Opis początkowy może być sygnaturą funkcji obiektu źródłowego lub nazwą atrybutu i oznacza źródło pochodzenia informacji. Brak opisu źródłowego lub docelowego oznacza, że cały obiekt jest odpowiednio źródłem lub przeznaczeniem informacji i od kontekstu zależy, jak informacja będzie pobierana lub zapisywana.



Rys. 5. Przepływ danych przez proces, aktywność i warunek



Rys. 6. Przykład określania kolejności wykonywania operacji: a) za pomocą wektora przepływu sterowania, b) za pomocą numeratorów

Przepływ danych skierowany do lub od ramki procesu oznacza przekazanie danych do lub od procesu (rys. 5). Nazwy przy wektorach przepływu są nazwami parametrów procesu. W ten sam sposób jest traktowana aktywność i warunek. Przy bloku aktywności dane są przekazywane przy każdej iteracji. W przypadku warunku dochodzi jeszcze konieczność oznaczenia wyjścia/wejścia wybieranego przy warunku spełnionym i przy niespełnionym. Do tego celu służy dodatkowy opis wektora.

Kolejność operacji często może zostać wyznaczona na podstawie przepływu danych. Są jednak sytuacje, gdy nie jest to możliwe (rys. 6). Wówczas można wykorzystać przepływ sterowania. Przepływ sterowania przedstawia kolejność wykonywania operacji. Wektor przepływu jest reprezentowany przez przerywaną strzałkę. Może łączyć procesy, aktywności, warunki, stany, struktury sterowania, wektory przepływu danych. Alternatywnym sposobem jest użycie tzw. numeratorów, tj. liczb porządkowych zakończonych dwukropkiem określających kolejność. Numeratory można umieścić przy wektorach lub przy procesach.

Przedstawiony tutaj opis notacji projektowej IMAC nie jest kompletny i nie wyczerpuje całości zagadnienia. Jest jednak wystarczający dla zrozumienia kolejnego rozdziału, w którym pokażemy sposób interpretacji diagramów interakcji podczas implementacji w kontekście metody IMAC.

4. IMPLEMENTACJA PRZEZ DIAGRAMY INTERAKCJI

Diagramy interakcji są w UML wykorzystywane do opisu przypadków użycia, opisu operacji lub mogą występować jako diagramy samodzielne opisujące pewne kluczowe fragmenty behawioryzmu systemu. W tym rozdziale przedstawiony zostanie sposób uszczegółowienia diagramów funkcjonalnych wykorzystywanych do opisu przypadków użycia tak, aby mogły zostać użyte podczas implementacji do wygenerowania funkcjonalnego kodu.

W UML zdefiniowane są dwie formy diagramów interakcji: diagramy kolaboracji i diagramy sekwencji. Ponieważ obie formy są semantycznie równoważne, zajmiemy się diagramami sekwencji, które ze względu na wprowadzenie wymiaru czasu są bardziej podatne na interpretację implementacyjną.

Diagramy sekwencji w UML zawierają symbole aktorów i obiektów składowych systemu, linie życia reprezentujące oś czasu z pokazanymi czasami skupienia sterowania (ang. *focus of control*) oraz wektory przepływu komunikatów między nimi. W IMAC diagramy sekwencji mogą być wczytane i wykorzystane wprost z drobną modyfikacją polegającą na zastąpieniu symboli obiektów ich ikonidami, a wektorów przepływu komunikatów wektorami przepływu danych. Linie życia obiektów mogą w notacji IMAC zostać uznane za specjalną formę wektorów przepływu sterowania, a czas skupienia sterowania za specjalną formę procesu.

Celem translacji diagramu sekwencji w diagram implementacji jest doprowadzenie go do takiego stopnia uszczegółowienia, aby można było za pomocą narzędzia do generacji kodu wytworzyć w pełni funkcjonalny kod źródłowy. Kod taki będzie można skompilować i uruchomić bez dalszej ingerencji projektanta. Diagramy sekwencji UML nie nadają się do wytworzenia takiego kodu m.in. z następujących przyczyn:

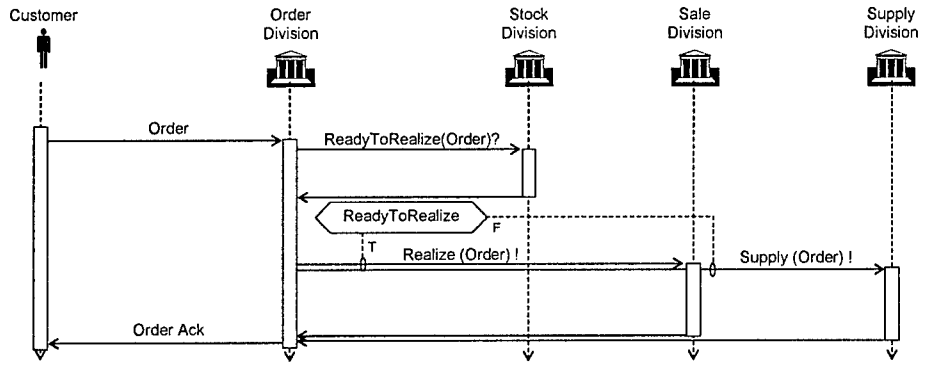
- Zawierają elementy modelowe nie posiadające bezpośrednich odpowiedników w dziedzinie oprogramowania (np. aktorzy).

- Opisują współdziałanie obiektów z perspektywy całego systemu, nie opisują operacji poszczególnych obiektów.
- Słabo wspierają opisanie iteracji, warunków, różnych ścieżek wykonania i innych szczegółów implementacji.

W dalszym ciągu pokażemy sposoby rozwiązania tych problemów w metodzie IMAC. Jako przykład rozważymy przypadek użycia Order Placement reprezentujący złożenie zamówienia przez klienta. Scenariusz tego przypadku użycia może zostać opisany następująco:

Klient składa zamówienie poprzez Internet. Dział zamówień sprawdza we współpracy z działem magazynowym, czy zamówienie może zostać zrealizowane w trybie natychmiastowym. Jeżeli tak, to przekazuje zamówienie do działu sprzedaży w celu realizacji i wysyła odpowiednią informację do klienta. Jeżeli nie, to do działu zaopatrzenia wysyła zamówienie wewnętrzne na brakujące towary i wysyła potwierdzenie zamówienia do klienta. Zamówienie zostanie zrealizowane po sprowadzeniu zamówionego towaru przez dział zaopatrzenia.

Diagram interakcji dla tego przypadku użycia przedstawia rys. 7. Znak zapytania po ReadyToRealize oznacza, że jest to zapytanie. Komunikat zwrotny jest odpowiedzią na to pytanie. Wykrzyknik po Realize i po Supply oznacza żądanie. Komunikat zwrotny po żądaniu nie niesie istotnej informacji, a jest jedynie sygnałem o zakończeniu realizacji żądania.

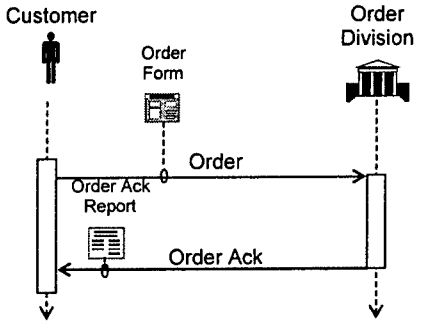


Rys.7. Diagram interakcji dla złożenia zamówienia

4.1. Implementacja aktorów

Aktor jest w UML modelowany za pomocą klasy stereotypu **actor**. Właściwości (atrybuty) aktora będą dla systemu informatycznego danymi pobieranymi od użytkownika i ew. przechowywanymi w bazie danych systemu. Do pobierania danych służą formularze. W UML nie ma bezpośrednio zdefiniowanego elementu „formularz", jest on definiowany w pewnych rozszerzeniach UML. W IMAC formularz jest reprezentowany przez klasę stereotypu **form**. Formularz może zostać umieszczony na drodze przepływu danych od użytkownika do obiektu składowego systemu (rys. 8). Właściwości formularza reprezentują pobierane dane. Powinny stanowić podzbiór sumy właściwości aktora (użytkownika +

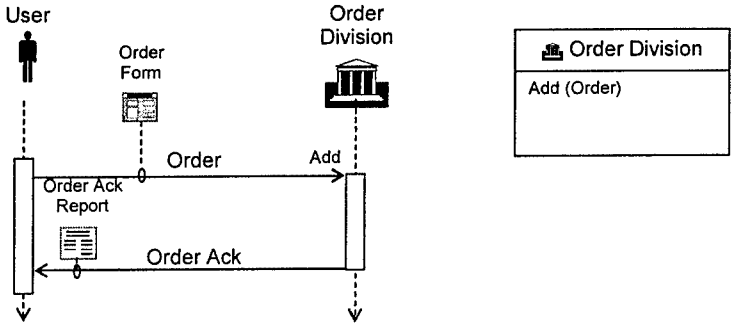
ew. klienta) i przekazywanej informacji. Podobnie dane przekazywane z systemu do użytkownika (klienta) są implementowane poprzez klasę stereotypu **report**.



Rys. 8. Użycie formularza i raportu do implementacji aktora

4.2. Implementacja przez operacje

W UML przepływ komunikatu reprezentuje najczęściej wykonanie operacji. Nazwa operacji wraz z ew. listą argumentów jest podawana w opisie wektora komunikatu. W IMAC wektor przepływu danych jest opisany nazwą przekazywanej informacji. Nazwa operacji obiektu docelowego, która będzie wywoływana dla przyjęcia tej informacji może zostać umieszczona w opisie końca wektora (rys. 9). Argumentem dla tej operacji będzie przesyłana informacja. Sygnatura operacji pojawi się w specyfikacji klasy docelowej.



Rys. 9. Oznaczenie operacji docelowej dla przyjęcia komunikatu

4.3. Szczegóły implementacyjne

Dla potrzeb implementacji trzeba jednak jeszcze ustalić wiele szczegółów, m.in.:

1. Czy formularz Order Form jest tworzony dla potrzeb tej interakcji, czy też jest istnieje jeden globalny dla całego systemu?
2. Jaka operacja odpowiada za pokazanie formularza na ekranie?
3. Jak jest reprezentowana informacja Order?
4. Jaka forma komunikacji zostanie zastosowana do wykonania operacji Add?

W odpowiedzi na pierwsze pytanie: jeżeli formularz (czy jakikolwiek inny obiekt) jest tworzony wyłącznie na potrzeby tej interakcji, to w jego zastrzeżeniach podaje się słowo kluczowe {**local**}. W przeciwnym wypadku obiekt będzie tworzony globalnie.

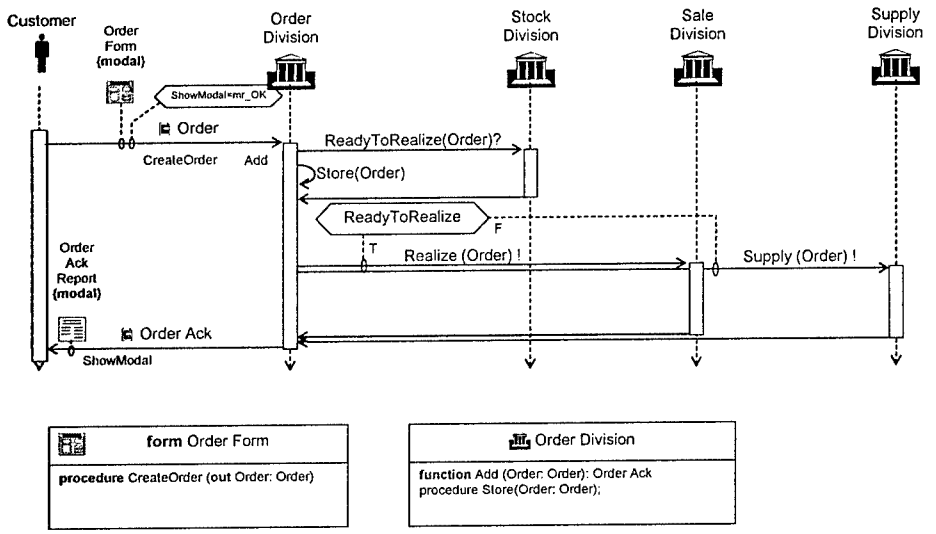
Aby odpowiedzieć na drugie pytanie trzeba ustalić sposób wyświetlania formularza. Formularz może być pokazywany na ekranie w sposób modalny lub niemodalny. Formularz modalny musi być wypełniony i zamknięty przed kontynuowaniem pracy. Formularz niemodalny umożliwia przełączanie się użytkownika pomiędzy różnymi oknami ekranu, w tym również rozpoczęcie innej interakcji zanim zostanie zakończona bieżąca. Jeżeli projektant zdecyduje się na modalny formularz, to w jego zastrzeżeniach podaje słowo kluczowe {**modal**}. Operację odpowiadającą za pokazanie formularza na ekranie ustala się w zależności od przyjętego sposobu wyświetlania oraz od środowiska implementacji (biblioteki komponentów implementacyjnych). Dla przykładu w bibliotece VCL firmy Borland za pokazanie niemodalne odpowiada procedura Show, a za pokazanie modalne funkcja ShowModal zwracająca wynik zależny od sposobu zamknięcia okna formularza.

W odpowiedzi na pytanie trzecie: informacja jest obiektem o nazwie Order klasy Order. Metoda IMAC umożliwia stosowanie tej samej nazwy dla obiektu i dla klasy. Odróżnienie obiektu od klasy jest możliwe poprzez kontekst. W przypadku niejednoznaczności nazwę klasy można poprzedzić słowem kluczowym **class**.

Trzeba ustalić czas życia obiektu Order. Na pewno zostaje utworzony przez formularz Order Form. Po zakończeniu interakcji nie jest już potrzebny, zatem jest to obiekt tymczasowy. W metodzie IMAC domyślnie obiekty wykorzystywane do przekazywania informacji wewnątrz interakcji są obiektami tymczasowymi. Jeśliby czas życia obiektu miał przekraczać czas trwania operacji, to taki obiekt powinien zostać oznaczony zastrzeżeniem {**new**}. Dla zaznaczenia, że obiekt jest tworzony przez operację CreateOrder nazwę tej operacji umieszczamy przy konektorze formularza.

Pytanie czwarte jest pytaniem o formę komunikacji dla operacji Add. Do wyboru jest forma proceduralna, synchroniczna i asynchroniczna. Forma proceduralna jest formą jednowątkową, wątek sterowania przechodzi od operacji wywołującej do operacji wywoływanej. Dane przekazywane są przez parametry, wynik operacji może być przekazany przez parametr wyjściowy albo przez wynik funkcji. Operacja proceduralna jest oznaczana przez stereotyp **procedure** albo **function**. Forma synchroniczna i asynchroniczna są formami wielowątkowymi. Polegają na przekazaniu komunikatu od obiektu źródłowego do obiektu docelowego. Różnią się tym, że w formie synchronicznej wątek sterowania obiektu źródłowego zostaje wstrzymany do czasu zakończenia obsługi komunikatu przez obiekt docelowy, zaś w formie asynchronicznej obiekt źródłowy kontynuuje wykonywanie wątku bez czekania na obsłużenie komunikatu. W obu formach przekazywanie danych odbywa się przez parametry komunikatu, przy czym w formie asynchronicznej wynik operacji musi być przekazany przez komunikat zwrotny. Operację dla komunikacji asynchronicznej i synchronicznej oznacza się stereotypem **message**. Wybór pomiędzy komunikacją synchroniczną a asynchroniczną zależy od układu komunikatów.

Te i inne szczegóły implementacyjne są zaznaczane na diagramie implementacyjnym, który składa się z diagramu sekwencji i diagramów statycznych pokazujących definicje operacji (rys. 10). Zwróćmy uwagę na operację Store, której zabrakło na poprzedniej wersji diagramu sekwencji, a która jest konieczna dla możliwości późniejszej realizacji zamówienia.



Rys. 10. Przykładowy diagram ze szczegółami implementacyjnymi

Dla przykładowego diagramu (rys. 10) zostanie wygenerowany następujący kod w języku IML:

```
var
  Order: Order;
  Order_Ack: Order_Ack;

if Order_Form.ShowModal=mr_OK then
 begin
  Order_Form.CreateOrder (Order);
  Order_Ack ← Order_Division.Add(Order);
  Order_Ack_Report.Order_Ack ← Order_Ack;
  Order_Ack_Report.ShowModal;
  free Order, Order_Ack;
end
...
function Order_Division.Add (Order: Order): Order_Ack;
begin
  send ReadyToRealize(Order) to Stock_Division;
  Store(Order);
  wait;
  if Order.ReadyToRealize then
    Sale_Division.Realize(Order)
  else
    Supply_Division.Supply(Order);
end
```

Język IML jest językiem pośrednim, który umożliwia prostą translację na język C++, C# czy Object Pascal. W praktyce projektant nie musi oglądać wygenerowanego kodu, ani tym bardziej w niego ingerować. Wszelkie szczegóły implementacyjne może umieścić na opracowywanym diagramie interakcji.

5. PODSUMOWANIE

W tej pracy pokazano, że metoda IMAC umożliwia doprowadzenie diagramów implementacyjnych do takiego poziomu szczegółowości, by można było z nich wygenerować funkcjonalny kod aplikacji. Umożliwia to odejście od dotychczasowej technologii *pisania* programów na rzecz ich *konstruowania* i może stanowić przełom w technologii informatycznej na miarę tego, który nastąpił przy zastąpieniu programowania w asemblerze programowaniem w językach wysokiego poziomu. Skutkiem tego konstruktor nie będzie analizował algorytmu instrukcja po instrukcji, lecz będzie tworzył rozwiązania konstrukcyjne i opisywał je na diagramach implementacyjnych. Zastosowanie jednolitej notacji w całym projekcie informatycznym umożliwi lepsze zrozumienie miedzy projektantem a konstruktorem i zapewni wyeliminowanie wielu potencjalnych źródeł błędów występujących w dotychczasowej metodologii wytwarzania oprogramowania.

BIBLIOGRAFIA

[1] *OMG Unified Modeling Language Specification*, version 1.5 March 2003, http://www.omg.org/docs/formal/03-03-01.pdf
[2] Dennis A., Haley Wixom B., Tegarden D.: *System Analysis & Design. An Object-Oriented Approach with UML*, John Wiley & Sons, Inc., USA 2001
[3] Borland Delphi 7.0 User's Guide, Borland, Scotts Valley, 2002
[4] Borland C++ Builder v.5.0 User's Guide, Borland, Scotts Valley, 2000
[5] Microsoft Visual Studio .NET Users' Guide, Microsoft 2002
[6] Bohm C., Jacopini G.: Flow Diagrams, Turing Machines and Languages with only Two Formation Rules, CACM, vol.9, no.5, May 1966
[7] Nassi I., Shneiderman B.: *Flowchart Techniques for Structured Programming*, SIGPLAN Notices, ACM, August 1973
[8] Chapin N.: *A New Format for Flowcharts*, Software-Practice and Experience, vol.4, no.4, 1974

INTEGRATION OF SOFTWARE MODELING AND IMPLEMENTATION IN IMAC

Summary

Present software engineering technology accepts lack of integrity between UML modeling and implementation by object oriented programming. IMAC uses unified graphic notation in the whole software engineering process. Through this programmers will become software constructors and will not write programs but will add implementation solutions to design diagrams and then generate fully functional program code.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Jerzy Nawrocki, Piotr Pawałowski, Marcin Werla

Instytut Informatyki, Politechnika Poznańska

HISTORIA REZULTATÓW TESTÓW JEDNOSTKOWYCH TWORZONA ZA POMOCĄ CRUISECONTROL I MAVEN

Streszczenie

W artykule przedstawiono aplikację, będącą rozszerzeniem narzędzia Maven, służącą do generowania i prezentowania historii rezultatów wykonań testów jednostkowych. Testy te wykonywane są automatycznie przez narzędzie CruiseControl zgodnie z założeniem ciągłej integracji, będącej jedną z podstawowych praktyk Programowanie Ekstremalnego (XP). Aplikacja powstała na Politechnice Poznańskiej jako część projektu mającego na celu wspomaganie zarządzania przedsięwzięciami realizowanymi w ramach Studia Rozwoju Oprogramowania (SDS).

1. WSTĘP

Obecnie na świecie jedną z najpopularniejszych lekkich metodyk programowania jest Programowanie Ekstremalne (XP). Proponuje ono wiele praktyk mających na celu polepszenie procesu wytwarzania oprogramowanie. Są to m.in. ciągła integracja powstającego oprogramowania oraz ciągłe jego testowanie. Programowanie Ekstremalne jest stosowane w ramach Studia Rozwoju Oprogramowania (SDS).

Praktyczne połączenie wspomnianych praktyk można uzyskać wykorzystując dwa narzędzia typu *open source*: CruiseControl [7] oraz Apache Maven [3]. Pierwsze z nich służy do sprawdzania z zadaną częstotliwością stanu repozytorium projektu i w przypadku wykrycia zmian wykonywaniu zdefiniowanych przez programistę czynności, takich jak np. wykonanie testów jednostkowych, a następnie prezentowanie ich na stronach WWW. Drugie zaś ma na celu ułatwienie wielu aspektów pracy nad rozwojem oprogramowania, m.in. takich jak projektowanie oprogramowania, zarządzanie konfiguracją, wymianę informacji między osobami związanymi z projektem, czy wreszcie zapewnianie jakości oprogramowania poprzez automatyczne inspekcje kodu i różnego rodzaju testy. Rezultaty jego działania przedstawiane są m.in. w postaci raportów, stanowiących część generowanej przez Mavena strony WWW projektu.

Obydwa narzędzia prezentują na generowanych przez siebie stronach WWW wyniki testów jednostkowych, jednak Maven generuje raport tylko z ostatniego wykonania testów, a CruiseControl prezentuje osobno raporty z testów wykonywanych podczas kolejnych integracji. Raporty te jednak nie mają żadnego wspólnego podsumowania, czy statystyk.

Z tego też powodu powstał pomysł stworzenia UnitTestHistory, modułu Mavena, umożliwiającego generowanie raportów podsumowujących wyniki testów jednostkowych wykonywanych podczas kolejnych integracji projektu. Moduł ten wykorzystując dane na

temat poszczególnych integracji, zbierane przez CruiseControla, dodaje do generowanej przez Mavena strony raport pokazujący, w jaki sposób przez czas trwania projektu przyrastała liczba testów w poszczególnych modułach oraz w całym projekcie, jaki był w poszczególnych próbach integracji stosunek testów wykonanych pomyślnie do całkowitej liczby testów, oraz jak zmieniał się czas wykonywania testów. UnitTestHistory powstał z myślą o projektach realizowanych w ramach SDS. Ma ułatwiać pracę kierownikom projektów i audytorom oraz motywować programistów do większej dbałości o jakość tworzonego oprogramowania.

2. STUDIO ROZWOJU OPROGRAMOWANIA

Studio Rozwoju Oprogramowania (SDS) jest prowadzone na Politechnice Poznańskiej od 1998 roku i ma służyć studentom do zdobycia doświadczenia w zakresie prowadzenia przedsięwzięć programistycznych. Co roku realizowanych jest 11 przedsięwzięć, w których bierze udział łącznie 88 studentów z III, IV i V roku informatyki. Przedsięwzięcie trwa dwa semestry. Zespoły studenckie są podzielone na dwie grupy, z których każda ma własnego opiekuna pełniącego rolę "wyższego kierownictwa" (ang. Senior Management, Senior Supplier). Ponadto każde przedsięwzięcie ma promotora dbającego, aby działania studentów III roku doprowadziły ich do skutecznej obrony pracy inżynierskiej (ich zadaniem jest projektowanie, implementacja i testowanie produktu). Każde z przedsięwzięć posiada też klienta zgłaszającego propozycję przedsięwzięcia. Studenci V roku występują w roli kontrolerów jakości, natomiast studenci IV roku pełnią role Kierownika Przedsięwzięcia i Kierownika Produktu. Przedsięwzięcia zarządzane są zgodnie z metodyką PRINCE 2 [11], która powstała w Wielkiej Brytanii i obecnie zdobywa coraz większą popularność na świecie. Programiści natomiast, pracują zgodnie z zaleceniami metodyki Programowania Ekstremalnego (XP), stosując się do zalecanych w niej praktyk, takich jak programowanie parami, automatyczne testy jednostkowe, czy ciągła integracja kodu.

3. CIĄGŁA INTEGRACJA KODU ŹRÓDŁOWEGO

3.1. Ciągła integracja jako jedna z praktyk XP

W Programowaniu Ekstremalnym wyróżnia się cztery podstawowe fazy rozwoju oprogramowania: planowanie, projektowanie, kodowanie i testowanie. Każda z nich ma swoje praktyki [1]. Jedna z praktyk dotyczących kodowania brzmi „Często integruj" (ang. *Integrate often*). Mówi ona, iż kod źródłowy rozwijanego oprogramowania powinien być jak najczęściej integrowany, przy czym owo „jak najczęściej" rozumiane jest jako okres kilku godzin lub – w maksymalnych przypadkach – okres jednego dnia.

Główne zalety płynące ze stosowania ciągłej integracji to m.in.:

- Brak sesji integracyjnych – po wdrożeniu ciągłej integracji nie ma już konieczności organizowania grupowych sesji integracyjnych. W trakcie takich sesji wszyscy programiści, którzy pracują w danym projekcie ręcznie integrują wprowadzone przez siebie zmiany starając się wspólnie doprowadzić kod źródłowy do stanu, w którym będzie się on kompilował i po uruchomieniu działał zgodnie z założeniami. Sesje te zastąpione są wieloma małymi i częstymi integracjami wykonywanymi przez pary programistów po każdej wprowadzonej przez nich zmianie.

- Zmniejszenie liczby błędów związanych z procesem integracji – wykonywanie częstych i małych integracji znacząco wpływa na zmniejszenie liczby błędów, powstałych w systemie w skutek samej integracji. Jest tak dlatego, iż osoba integrująca kod w krótkim okresie po wprowadzeniu zmian (jak napisano wyżej, według zaleceń jest to maksimum jeden dzień), dużo lepiej pamięta, jakie dokładnie zmiany zostały wprowadzone i na jakie inne części oprogramowania zmiany te mogą mieć wpływ. W przypadku, gdy integracje wykonywane są dużo rzadziej, np. raz na tydzień, istnieje duże prawdopodobieństwo, iż do czasu integracji programista zdąży zapomnieć o części zmian.
- Lepsza komunikacja w zespole programistów – wraz z przeprowadzeniem każdej integracji programista, który ją wykonuje dowiaduje się o zmianach, które w międzyczasie wprowadzili pozostali członkowie zespołu. Jeżeli integracje są częste, zwiększa się wiedza wszystkich członków zespołu programistycznego na temat zmian, które są wykonywane w projekcie i w ten sposób spada prawdopodobieństwo duplikowania pewnej funkcjonalności systemu przez pracujących niezależnie od siebie programistów. Mniejsze są również szanse na to, że np. w skutek niezrozumienia dokumentacji przez jednego z programistów część systemu będzie rozwijana niezgodnie ze specyfikacją.
- Dostarczanie aktualnej postaci wykonywalnej rozwijanego produktu – efektem każdej udanej integracji jest postać wykonywalna rozwijanego oprogramowania. Wersja ta może być przekazana do zespołu testującego oprogramowanie w celu wykonania testów akceptacyjnych, czy też zweryfikowania zgodności rozwijanego oprogramowania ze specyfikacją w dowolny inny sposób.

Za udaną integrację Fowler i Foemmel uznają taką, w której uda się wykonać cztery następujące czynności [2]:

- Z systemu zarządzania wersjami zostanie pobrany aktualny kod źródłowy tworzonego oprogramowania.
- Cały kod źródłowy zostanie skompilowany.
- Na podstawie wyników kompilacji zostaną utworzone wersje wykonywalne artefaktów.
- Na powstałej w poprzednich trzech krokach wersji systemu zostaną uruchomione wszystkie zdefiniowane testy i każdy z uruchomionych testów zakończy się sukcesem.

Przy takim podejściu kluczowe czynniki niezbędne do skutecznego stosowania ciągłej integracji to posiadanie jednego centralnego repozytorium kodu źródłowego, tworzenie automatycznych testów jednostkowych oraz korzystanie ze zautomatyzowanych skryptów do kompilowania kodu źródłowego, uruchamiania testów oraz tworzenia postaci wykonywalnej oprogramowania. W obecnej chwili jednym z najlepszych systemów do automatyzowania wyżej wymienionych czynności jest Apache Maven.

3.2. Automatyzacja podstawowych czynności programisty – Apache Maven

Maven [3] jest to darmowe narzędzie wspierające na różnych płaszczyznach pracę nad oprogramowaniem tworzonym w Javie. Sam Maven również został napisany w Javie, dzięki czemu jest w pełni przenośny i dostępne na wielu platformach. Autorami Mavena są programiści z fundacji Apache, a licencja, na której jest on rozpowszechniany to Apache Software License ver.1.1 [4].

Główne funkcje Mavena to automatyzowanie podstawowych zadań programistycznych oraz ułatwienie zarządzania projektem i nadzorowanie jego jakości. Maven wyko-

rzystuje inne narzędzie fundacji Apache – Anta [5] i przez to daje nam pełną jego funkcjonalność, czyli między innymi kompilowanie kodu źródłowego, uruchamianie testów jednostkowych, tworzenie różnego rodzaju artefaktów, ale także przesyłanie plików przy użyciu narzędzi FTP czy SCP lub na przykład wysyłanie wiadomości e-mail. Dodatkowo Maven zwiększa możliwości oferowane przez Anta o pełną funkcjonalność skryptów Jelly [6], które oferują np. zaawansowane możliwości transformacji XML-a.

Dzięki tym możliwościom oraz rozbudowanemu mechanizmowi rozszerzeń Maven m.in. generuje stronę WWW projektu, w którym jest używany. Strona ta zawiera podstawowe informacje o projekcie, takie jak listę członków projektu czy listę zewnętrznych bibliotek używanych w projekcie wraz z numerami ich wersji, a także wiele raportów z informacjami uzyskanymi na skutek automatycznej analizy kodu źródłowego. Raporty te przedstawiają między innymi podstawowe metryki kodu (np. NCSS i CCN), ale także wyniki poszukiwania duplikatów w kodzie, listę niezgodności z przyjętym standardem kodowania, zestawianie wyników i czasów trwania testów jednostkowych czy też szczegółowe informacje na temat pokrycia kodu źródłowego przez testy jednostkowe.

Ze względu na bardzo rozbudowaną funkcjonalność Maven doskonale sprawdza się we wspomaganiu rozwoju oprogramowania i może być z powodzeniem stosowany do automatyzowania procesu ciągłej integracji. Kolejnym narzędziem, które w połączeniu z Maven-em w pełni automatyzuje proces integracji jest CruiseControl.

3.3. Ciągła integracja – CruiseControl

CruiseControl [7], podobnie jak Maven, jest darmowym, napisanym w Javie programem. Jego głównym zadaniem jest ułatwienie i zautomatyzowanie omawianego wcześniej procesu ciągłej integracji. CruiseControl składa się z dwóch części: z programu realizującego proces integracji oraz z aplikacji WWW służącej do przeglądania wyników poszczególnych integracji. Kluczowa jest oczywiście część realizująca regularne integracje.

Część ta raz uruchomiona regularnie sprawdza stan wskazanego repozytorium kodu źródłowego. Okresowość sprawdzania może być zdefiniowana albo poprzez względny okres czasu pomiędzy kolejnymi sprawdzeniami repozytorium (np. jedna godzina) lub poprzez podanie konkretnego momentu czasu, w którym ma następować sprawdzenie repozytorium (np. codziennie o godzinie 18.00).

Jeżeli podczas sprawdzania repozytorium zostaną wykryte wprowadzone przez programistów zmiany, to rozpoczynany jest proces integracji. Przebieg procesu integracji definiowany jest w skrypcie automatyzującym integrację i zazwyczaj jest on zbliżony do procesu opisanego w punkcie 3.1. W obecnej wersji (2.1.5) CruiseControl pozwala na wykorzystanie skryptów Ant-a oraz Mavena. Po zakończeniu integracji, odpowiednio skonfigurowany CruiseControl rozsyła e-mailem informacje o tym, czy integracja powiodła się czy nie. Jeżeli integracja nie powiedzie się, to przy następnym sprawdzeniu repozytorium rozpocznie się próba ponownej integracji, nawet jeżeli nie wprowadzono żadnych zmian. To pozornie dziwne postępowanie ma swoje uzasadnienie w sytuacji, kiedy integracja nie powiedzie się nie dlatego, że kod źródłowy jest błędny lecz np. z powodu czasowej niedostępności łącza internetowego niezbędnego do wykonania części testów systemu. Wtedy ponowna integracja wykonana po pewnym czasie ma szanse zakończyć się sukcesem nawet, jeżeli nie wprowadzono żadnych zmian w kodzie tworzonego oprogramowania.

Przebieg każdej integracji wykonywanej przez CruiseControla jest zapisywany do pliku w formacie XML. Pliki te są następnie wykorzystywane przez wspomnianą wcześniej aplikację WWW do prezentowania raportów. Raport z każdej integracji zawiera wszystkie

komunikaty, które wygenerował skrypt integrujący, a także zestawienie przebiegu testów jednostkowych. Niestety brakuje tu raportów pokazujących historię projektu czy też podsumowania lub analizy przeprowadzonych integracji.

4. HISTORYCZNIE ZINTEGROWANE RAPORTY TESTOWANIA - UNIT TEST HISTORY

4.1. Motywacja

Studio Rozwoju Oprogramowania, jak każdy dobry producent oprogramowania, dąży do systematycznego polepszania swoich procesów. Dlatego od dwóch lat realizowane są projekty takie, jak na przykład e-SDS [8], mające na celu stworzenie oprogramowania do użytku wewnętrznego SDS. Jednym z takich projektów jest UnitTestHistory, moduł narzędzia Maven, stworzony do generowania raportów podsumowujących wyniki testów jednostkowych podczas kolejnych integracji projektu. Raporty te tworzone są na podstawie plików logu programu CruiseControl. Moduł pozwala kierownictwu przedsięwzięć realizowanych w ramach SDS na regularną kontrolę postępów prac programistycznych, przez obserwację zmian w liczbie testów jednostkowych. Dostęp do tych raportów jest prosty, ponieważ są one automatycznie publikowane na stronie WWW, każdego przedsięwzięcia. Również programiści chętnie odwiedzają stronę z raportami, aby zobaczyć czy poszczególne integracje powiodły się i czy nie powstały jakieś nowe błędy.

4.2. Funkcjonalność

UnitTestHistory jest aplikacją napisaną w języku programowania Java i wykorzystuje różne darmowe biblioteki dostępne dla tego języka. Pliki logów CrusieControl są zapisane w postaci XML – do pozyskania danych z tych plików użyto biblioteki Digester [9]. Pozwala ona w wygodny sposób pobrać dane z pliku XML i przedstawić je jako zbiór obiektów w Javie. Biblioteka ta wykorzystywana jest również do wczytania pliku konfiguracyjnego UnitTestHistory, o którym zostanie omówiony w dalszej części artykułu. Na podstawie danych z pliku konfiguracyjnego i logów CruiseControl, przy pomocy mechanizmów Mavena generowana jest strona internetowa z wybranymi przez użytkownika wykresami związanymi z historią wykonywanych testów jednostkowych. Do generowania wykresów wykorzystywana jest biblioteka JFreeChart [10]. Umożliwia ona generowanie wielu rodzajów wykresów i zapisywanie ich w postaci plików graficznych. Obecna implementacja UnitTestHistory wykorzystuje cztery rodzajów wykresów: wykres skumulowany kolumnowy (StackedBarChart), wykres skumulowany kolumnowy 3D (StackedBarChart3D), wykres przebiegu czasowego (TimeSeriesChart) oraz wykres skumulowany warstwowy (StackedAreaXYChart). W logach CruiseControl dostępne są następujące rodzaje danych: liczba testów zakończonych pomyślnie (NoOfSuccess), liczba testów błędnie wykonanych (NoOfErrors), liczba testów zakończona niepomyślnie (NoOfFailures), całkowita liczba testów (NoOfTestCases), czas wykonania testów (TestCasesTime), czas przygotowania systemu do testów (TestFrameworkTime) oraz całkowity czas testowania (TotalTime). To, który rodzaj wykresu zostanie wykorzystany do prezentacji poszczególnych rodzajów danych, zależy wyłącznie od użytkownika.

4.3. Konfiguracja

Konfigurowanie UnitTestHistory nie wymaga od użytkownika dużego wysiłku. Pierwszy krok to standardowa instalacja UnitTestHistory jako plugin-a Mavena, czyli po prostu skopiowanie wersji dystrybucyjnej do podkatalogu plugins, katalogu głównego instalacji Mavena. Następnie w pliku „project.xml", w sekcji <reports> należy dodać wpis włączający generowanie raportu UnitTestHistory:

```
<reports>
        ........
        <report>maven-unittesthistory-plugin</report>
</reports>
```

Kolejna czynność to zdefiniowanie w pliku „project.properties" lub „build.properties" dwóch parametrów. Pierwszy z nich wskazuje katalog z logami CruiseControl-a:

```
unittesthistory.cclogs = /home/CruiseControl/logs
```

a drugi określa położenie pliku konfiguracyjnego dla UnitTestHistory:

```
unittesthistory.charts.configuration=/home/root/project/charts.xml
```

przy czym drugi z parametrów jest opcjonalny i służy do nadpisania standardowego zestawu generowanych wykresów.

Zestaw ten definiowany jest w osobnym pliku konfiguracyjnym o domyślnej nazwie „charts.xml". Pierwszą rzeczą, którą należy zdefiniować, jest klasa Javy, która odpowiedzialna będzie za pobranie danych. Aktualnie istnieje tylko jedna taka klasa, a zdefiniowanie jej użycia wygląda następująco:

```
<dataSourceClass>unittesthistory.loganalyzer.UnitTestHistory</dataSourceClass>
```

Następnie występuje dowolna liczba definicji wykresów. Pojedyncza definicja wykresu przedstawiona jest poniżej. Jako atrybuty podano rozmiary wykresu, kolor tła i typ wykresu (jeden spośród wspomnianych wcześniej czterech dostępnych):

```
<chart width="700" height="600" background="white" type="StackedAreaXYChart">
```

Dla każdego definiowanego wykresu należy podać tytuł, jaki ma się znaleźć nad wykresem, opisy osi oraz dłuższy opis widoczny pod wykresem:

```
<title>Number of tests per integration - whole project</title>
<categoryAxisTitle>Time</categoryAxisTitle>
<seriesAxisTitle>Number of tests</seriesAxisTitle>
<description>
        <![CDATA[
                This chart shows the total number of tests per
                integration.
        ]]>
</description>
```

Konieczne jest określenie zakresu czasowego, z którego dane mają być przedstawione na wykresie:

```
<categories from="first" to="last" period="7"/>
```

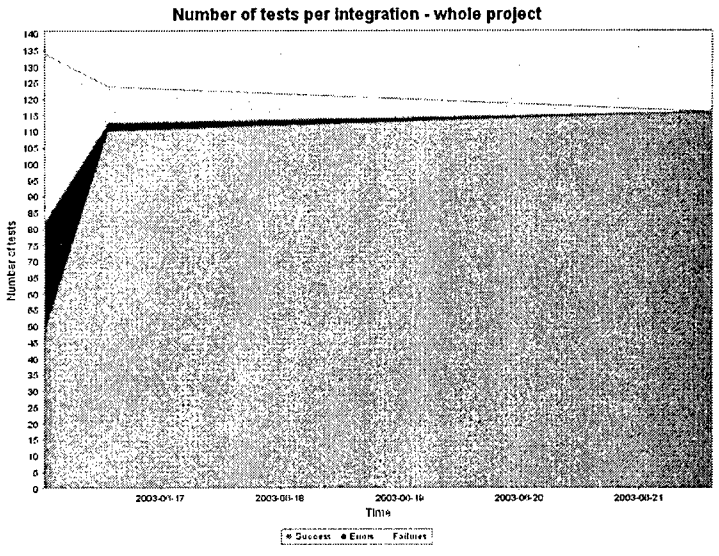
Jako wartości atrybutów „from" oraz „to" można podać datę rozpoczęcia i datę zakończenia przedziału czasowego w formacie YYYY-MM-DD, albo użyć słów kluczowych zdefiniowanych w programie: „first" lub „last", oznaczających odpowiednio pierwszy i ostatni dzień, z którego dostępne są dane. Można też zamiast atrybutów „from" i „to" użyć „latest" – spowoduje to wykorzystanie danych z ostatnich n dni, gdzie n to wartość atrybutu „latest". Bardzo ważny jest również parametr „period", który oznacza odstęp w dniach pomiędzy integracjami, które mają być brane pod uwagę. Jest to pomocne przy generowaniu wykresu dla długiego odcinka czasu, w sytuacji gdy nie jest ważne jak przebiegała każda integracja, lecz tylko to, jak zmieniały się pewne wartości (np. liczba testów) z dokładnością do tygodnia.

Ostatnim krokiem jest zdefiniowanie jakie serie danych mają zostać wzięte do wykresu.

```
<series name="Success" source="NoOfSuccess" color="green"/>
<series name="Errors" source="NoOfErrors" color="red"/>
<series name="Failures" source="NoOfFailures" color="yellow"/>
```

Określa się tu nazwę serii, typ danych oraz kolor w jakim poszczególna seria ma zostać przedstawiona na wykresie.

W taki sam sposób możemy zdefiniować dowolną liczbę wykresów. Gdy wykonamy wszystkie te kroki, to przy następnej generacji strony za pomocą narzędzia Maven, pojawi się raport UnitTestHistory. Na rysunku 1 przedstawiono wykres wygenerowany na podstawie powyższej konfiguracji.



Rys. 1. Przykładowy wykres UnitTestHistory

5. ZAKOŃCZENIE

UnitTestHistory jest obecnie wykorzystywany przez wiele projektów prowadzonych w ramach Studia Rozwoju Oprogramowania. Narzędzie to ułatwiło w znaczący sposób kontrolę nad procesem ciągłej integracji. Pozwoliło również kierownictwu przedsięwzięcia lepiej oceniać postępy prac programistycznych. W najbliższej przyszłości planowany jest rozwój dalszych narzędzi wspomagających SDS. Będą one służyły zbieraniu różnych metryk związanych z procesem wytwarzania oprogramowania. Wersja pilotowa UnitTestHistory została ukończona i przekazana do użytku zespołom realizującym przedsięwzięcia w ramach SDS. Publiczne udostępnienie narzędzia planowane jest po ukończeniu etapu testowania na Politechnice Poznańskiej.

BIBLIOGRAFIA

[1] *The Rules and Practices of Extreme Programming.* http://www.extremeprogramming.org/rules.html
[2] Fowler M., Foemmel M.: *Continuous Integration.* http://www.martinfowler.com/articles/continuousIntegration.html
[3] *Maven Project Homepage.* http://maven.apache.org/
[4] *The Apache Software License.* http://www.apache.org/licenses/LICENSE-1.1
[5] *Ant Project Homepage.* http://ant.apache.org/
[6] *Jelly Project Homepage.* http://jakarta.apache.org/commons/jelly/
[7] *CruiseControl Project Homepage.* http://cruisecontrol.sourceforge.net/
[8] Nawrocki J., Pawałowski P., Pospiech K.: *System Internetowy Wspomagający Wytwarzanie Oprogramowania Metodą Prince 2*, WETI PG, 2003
[9] *Jakarta Commons Homepage* http://jakarta.apache.org/commons/index.html
[10] *JFreeChart Homepage* http://www.jfree.org/jfreechart/index.html
[11] CCTA, *Managing Successful Projects with PRINCE 2*, The Stationary Office, London, 2002

CREATING UNIT TESTING RESULTS HISTORY USING CRUISECONTROL AND MAVEN

Summary

Nowadays one of the most popular software development methodologies is Extreme Programming (XP). One of the key XP practices is Continuous Integration. Because integration process is done very often, an automation of Integration Unit Testing is needed.
CruiseControl and Maven are tools that help programmers with Integration Unit Testing. The first of them monitors the state of the Version Management System repository, and when a change in the system code is detected, it checkouts the code, compiles it and runs Unit Tests. The information gained from tests is stored in log files. The second tool, Maven, is an application developed to help programmers with many common tasks including compiling the code, connecting to a Version Management System or copying files to specified locations. It also allows generating many reports showing different aspects of code quality, as automatically detected common programming errors, nonconformance to coding standards or status of last Unit Tests execution. Unfortunately there is no report showing the history of Unit Testing during the whole project.
UnitTestHistory was developed at the Institute of Computer Science at the Poznań University of Technology to allow automatic generation of test reports showing test results in a specified period of time (from the historical perspective). It is a Maven plug-in that can read CruiseControl log files. A UnitTestHistory report consists of charts showing different aspects of tests, such as number of successful tests, test errors, and failed tests in each integration or total testing time. A user can determine what charts are needed (using an XML configuration file).

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Rafał Sworowski, Henryk Lasota

Katedra Systemów Elektroniki Morskiej, Politechnika Gdańska

BADANIE WPŁYWU KONSTRUKCJI MODELU WIELOAGENTOWEGO NA JEGO ZDOLNOŚĆ PREDYKCYJNĄ

Streszczenie

Artykuł przedstawia metodykę i wyniki badań wpływu parametrów modelu wieloagentowego MAS na efektywność jego działania w zakresie symulacji rozwoju rynków IT, na przykładzie rozwoju rynku telefonii komórkowej tworzonego przez trzech konkurencyjnych operatorów. W symulatorze MAS zastosowano, dla określonego modelu popytowego, trzy odmienne mechanizmy aktywacji agentów, prowadzące do ustalania się różnego typu stanów dynamicznej równowagi rynkowej. Oceniono zdolności predykcyjne modelu, odnosząc wyniki do danych z wybranego rynku, oraz jego wydajność obliczeniową w zależności od liczby agentów i sposobu ich aktywacji.

1. WSTĘP

W badaniach będących przedmiotem niniejszego opracowania wykorzystano dostępne informacje ilościowe opisujące proces kształtowania się rynku telefonii komórkowej od momentu wprowadzenia jej na rynek do etapu jego nasycania się, który obserwujemy obecnie. Podjęto próbę odtworzenia w warunkach laboratoryjnych procesu oddolnej budowy rynku, na którym obecna jest stała liczba IAP-operatorów (3) oraz różna liczba IAP-klientów. W tym celu zastosowano koncepcję systemów wieloagentowych (ang. *Multiagent Systems – MAS*), zaś motorem symulacji uczyniono popyt generowany przez IAP reprezentujące klientów, pozostający w ścisłym związku z powiązaniami pomiędzy nimi a ich najbliższym sąsiedztwem, a także z doświadczeniem zdobytym w trakcie korzystania z usług IAP-operatorów. Czynnikiem wpływającym na ewolucję stanu IAP-klientów jest użyteczność usług komunikowana przez agenty symbolizujące operatorów. Szczegółowo mechanizmy kształtujące rynek wymiany opisano w [1].

Obliczeniowa ekonomia oparta na systemach wieloagentowych (ang. *Agent-based Computational Economics*) stanowi odłam nurtu modelowania systemów na kanwie teorii sterowania optymalnego. Swoimi początkami sięga połowy dwudziestego wieku, kiedy to laureat Nagrody Nobla Herbert Simon stworzył imitację systemu podejmowania decyzji.

Zbiór współdziałających inteligentnych jednostek (*Inteligentne Agenty Programowe* – ang. *Intelligent Agent Programs*) tworzy ewoluujący system, w którym zmiany powstające w wyniku lokalnych interakcji dają ująć się *ex post* w reguły makroskopowe, modyfikujące

z kolei ponownie relacje w skali mikro. Z uwagi na budowę obiektową, systemy MAS są stosunkowo proste w implementacji i efektywne w działaniu.

Artykuł jest próbą wykazania przydatności podejścia MAS do modelowania zjawisk gospodarczych, na przykładzie rynku telefonii mobilnej. Przebadano wydajność obliczeniową modelu oraz wrażliwość generowanych wyników na sposób jego implementacji. Oceniono zdolności predykcyjne modelu w zależności od liczby agentów, sposobu ich aktywacji.

2. STRUKTURA PROCESU MODELOWANIA

W odniesieniu do systemów deterministycznych adekwatna jest definicja modelu, przedstawiona w [2]: *„model jest pewnym wzorcem metodologicznym (paradygmatem), który pełni następujące funkcje:*

- *jest podstawą ustalania rozmiaru (stopnia), w jakim działanie jest sterowane, czyli w jaki sposób i na ile umożliwia ono realizację zamierzeń badawczych,*
- *wraz z odpowiednią teorią daje podstawę skorygowania wyników osiąganych za pomocą różnych technik,*
- *ujawnia, jakiego rodzaju wiedza jest potrzebna celem dostosowania do niego uzyskanych wyników."*

Zjawiska zachodzące w systemach technicznych odzwierciedla się w abstrakcjach wraz z odpowiednimi sekwencjami sterowań optymalnych $\{u^*; t = 0, 1, 2...N\text{-}1\}$ opisanych regułą decyzyjną $\gamma(t)$:

$$\gamma(t) = P_t \rightarrow u^*_t, \tag{2.1}$$

gdzie: $P_t = (y_t, u^*_{t-1})$ – prawdopodobieństwo uzyskania wyniku y_t w przypadku zastosowania strategii u^*_{t-1} w okresie $t\text{-}1$,

$u^*_{t-1} = (u^*_0, u^*_1, \ldots, u^*_{t-1})$ – zbiór sterowań optymalnych z okresów $t = 0, \ldots, t\text{-}1$.

W przypadku systemów ekonomicznych powyższa definicja sprawdza się w sposób ograniczony, co wiąże się z występowaniem w nich licznych elementów zachowujących się „racjonalnie". Idea „racjonalnych oczekiwań" sformułowana w 1961 roku przez Mutha polega na wprowadzeniu równań opisujących zachowanie się podmiotów gospodarujących, zgodnie z ich indywidualnym kryterium optymalności, co pozostaje w zgodzie z koncepcją MAS. Rozwiązanie tak postawionego problemu poszukiwania rozwiązania optymalnego można utożsamiać z tzw. punktem równowagi nieantagonistycznej występującym w grze dynamicznej według Nasha, którego położenie zależy m.in. od zbioru informacji dostępnych dla każdego z uczestników gry, od zobowiązania ich do przestrzegania reguł związanych z realizowaną strategią, a także od ich nieuświadomionego wpływu na stan innych uczestników gry. Badanie tak zdefiniowanego modelu polega na sprawdzaniu adekwatności jego elementów, w szczególności:

- dynamicznego modelu świata (w wersji dyskretnej),
- wskaźnika jakości sterowania dla każdego decydenta,
- struktury informacji,

do warunków, w których jest on stosowany.

3. MODELOWANIE Z WYKORZYSTANIEM MAS

Wiedza o konstruowaniu systemów MAS jest dziś stosunkowo szeroka i dobrze ugruntowana (Epstein i Axtell [3], Chen i Yeh [4], Axtell [5], Arthur [6]). Jak sugeruje Axtell [5], budowa pojedynczego IAP powinna czerpać z teorii metodologicznego indywidualizmu, włączając do niej zarówno modele zachowania konsumenckiego, jak i teorię generalnej równowagi w sensie Nasha. Istotne rozbieżności istnieją jednakże w zakresie sposobu reprezentacji firm w modelach MAS. Zgodnie z teorią generalnego stanu równowagi, firma jest stabilną koalicją IAP, podczas gdy modele szkoły neoklasycznej traktują ją jako pojedynczego racjonalnego aktora. Z uwagi na prostotę implementacyjną, koncepcja neoklasyczna znalazła szersze praktyczne zastosowanie. W oparciu o nią powstała idea modelu ekonomii opartego na inteligentnych agentach opisana przez Lane'a w 1993, a rozwinięta w 1997 przez Basu i Pryora. Podejście to zastosowano również w [1] uzyskując zadowalające rezultaty.

Zgodnie z dostępnymi w literaturze wynikami prowadzonych doświadczeń, najlepszym sposobem implementacji modelu wiedzy agenta jest ograniczenie jego racjonalności (ang. *Boundedly Rational Agent*)[1]. Tak zdefiniowany agent nie posiada pełnej informacji o otaczającym świecie i jest uwikłany w sieć powiązań socjalnych (lokalne interakcje o ograniczonym zasięgu). Poszukując aktywnie własnego lokalnego optimum, IAP rekalkulując adaptuje się do sytuacji w otoczeniu. W ten sposób stan systemu oscyluje wokół stanu równowagi zgodnego z koncepcją Nasha, przechodząc przez liczne stany równowagi niestabilnej [5],[7].

Kolejnym fenomenem wymagającym krótkiego wyjaśnienia jest zjawisko uniwersalności znane od dawna fizykom, polegające na tym, że w określonych systemach fizycznych zmiany zachowania elementów składowych modelu nie prowadzą do zmian jego zachowania w skali makro. Możliwym wyjaśnieniem tego zjawiska jest fakt, że w każdym kroku symulacji zachodzą nowe okoliczności, a ilość możliwych struktur oraz ich dynamika jest tak duża, że dokonanie optymalnego wyboru jest oczekiwaniem nierealistycznym. Przykładem z kręgu modeli MAS obrazującym ideę uniwersalności jest praca Nagela i Rasmussena 1994 (por. [3]) – model MAS ruchu ulicznego[2].

4. IMPLEMENTACJA MODELU RYNKU TELEFONII MOBILNEJ

Obiekty stanowiące reprezentację IAP na etapie implementacji, to fragmenty pamięci przechowujące zarówno zestawy danych (atrybuty opisujące poszczególne agenty), jak i funkcje służące modyfikowaniu tych danych (tzw. metody). W analizowanym modelu zaimplementowano dwie klasy IAP: IAP-klientów oraz IAP-operatorów (por. [9]), których schematyczne porównanie przedstawia tabela 1. Wartości parametrów charakteryzujących obiekty typu IAP-klient nadawane są zgodnie z rozkładem normalnym, wokół wartości średnich podanych w tabeli. Dla obiektów typu IAP-operator parametry dynamicznie zmieniają się w zależności od sytuacji, od podanych wartości początkowych, z założonym krokiem.

---

[1] W [3] przedstawiono modele wielkości firm oraz współczynnika przyrostu ich wielkości z wykorzystaniem agentów racjonalnych i o ograniczonej racjonalności – agenty racjonalne zwróciły niesatysfakcjonujące wyniki.

[2] Konkluzją jest stwierdzenie, że zachowania istotne dla tworzenia się korków są niezależne od działań pojedynczych IAP, co oznacza, że bez względu na ich styl jazdy będą one podróżowały przez zbliżony czas.

Tabela 1

Zestawienie porównawcze atrybutów i funkcji obiektów typu IAP-klient oraz IAP-operator, wraz z zastosowanymi wartościami parametrów

OBIEKT IAP-klient	Wartość parametru	OBIEKT IAP-operator	Wartość parametru
Podatność marketingowa; Współczynnik *learning by doing*; Współczynnik zapominania; Użyteczność minimalna; Użyteczność min dla rezygnacji; Oddziaływani PR;	0,6 0,33 0,5 1,3 1,7 0,6	Cena oferty; Poziom wydatków Public-relations; Poziom rentowności; Dostęp do kapitału; Okres związania z siecią	125 0,44 nn nieogr. 12 mies.
FUNKCJA inicjalizacja; FUNKCJA ocena sąsiadów; FUNKCJA oblicz użyteczność; FUNKCJA porównaj użyteczność; FUNKCJA zmień stan; FUNKCJA rysuj;		FUNKCJA modufikuj wydatki PR; FUNKCJA modyfikuj cenę; FUNKCJA komunikuj zmianę; FUNKCJA licz statystyki;	5% 2,5%

Źródło: Opracowanie własne.

Kluczem do modelu MAS są metody określające zachowanie się agentów, które są identyczne dla całej ich klasy i występują w całym programie sterującym zachowaniem się modelu tylko jeden raz[3]. Z uwagi na budowę obiektową, model agencki możliwy jest do wyspecyfikowania w mniej niż 1 000 liniach kodu C/C++ lub innego języka uniwersalnego, lub nawet w zaledwie 100 liniach kodu dedykowanych środowisk dla symulacji MAS (np. SWARM lub Ascape).

Na działanie modelu MAS wpływ mają następujące kluczowe elementy:

- liczba agentów,
- sposób ich aktywacji,
- liczba i sposób realizacji połączeń pomiędzy nimi.

Z oczywistych powodów model MAS posiada doskonałą skalowalność – zwiększenie liczby agentów poddawanych symulacji nie wymaga zazwyczaj wprowadzania zmian w kodzie programu, nie pozostaje jednak bez wpływu na wydajność modelu.

W niniejszych badaniach aktywacji agentów dokonano na trzy sposoby: synchronicznie (jednorodnie) – gdy każdy agent jest aktywowany jeden raz w danym kroku czasowym i zawsze w tej samej kolejności, asynchronicznie – w losowych odstępach czasowych, lub asynchronicznie – w każdym kroku czasowym (aktywacja w pełni asynchroniczna).

Jak twierdzi Axtell [3], zmiana porządku aktywacji agentów w sposób losowy w kolejnych krokach czasowych t, pozwala uniknąć powstawania artefaktów (ang. *artifacts*) spowodowanych przypadkowymi korelacjami pomiędzy agentami, nie wyspecyfikowanymi w kodzie programu warunkującego ich działanie. Obecności artefaktów należy się spodziewać, jeśli niewielkie zmiany w kodzie prowadzą do znacznych różnic w danych generowanych przez model. Z tego powodu celowym może być nadanie każdemu agentowi własnego generatora losowego odpowiadającego za jego aktywację.

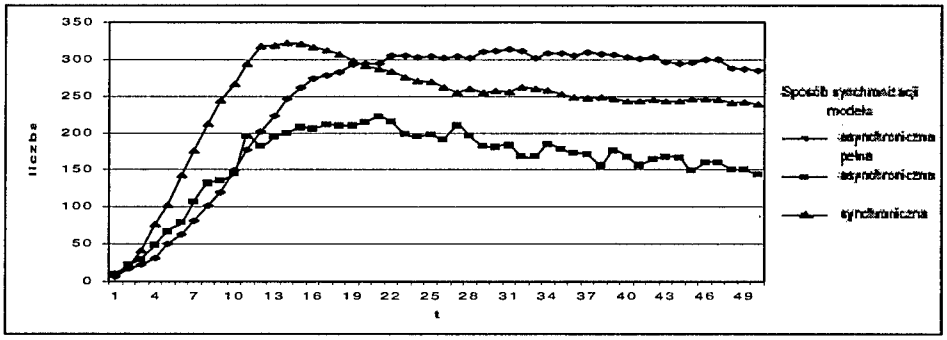
Zastosowany algorytm synchroniczny wybierał kolejne pozycje w macierzy agentów, dzięki czemu każdy agent był przetwarzany za każdym razem w tej samej kolejności. Aktywacji asynchronicznej dokonywano z kolei w oparciu o procedurę losowania ze zwracaniem. W przypadku „niepełnym", algorytm wybierał losowo agenta do przetworzenia, sprawdzając czy dany agent był już przetwarzany w danym kroku czasowym. Przy pozy-

[3] Bliższą charakterystykę funkcji wykorzystywanych przez obie grupy agentów przedstawiono w [1].

tywnej odpowiedzi na powyższe pytanie, losowano kolejnego IAP, powtarzając losowanie tyle razy, ile było agentów tworzących symulator. Dzięki temu, część z nich nie podlegała przetwarzaniu w danym kroku *t*. Aktywację asynchroniczną pełną zrealizowano podobnie – przed kolejnym krokiem symulacji, każdemu IAP przypisywano losowy i niepowtarzalny numer określający jego miejsce w kolejce do przetworzenia, po czym był on przetwarzany zgodnie z ustaloną kolejnością.

5. WYNIKI EKSPERYMENTÓW

W wyniku przeprowadzonych eksperymentów potwierdzono skuteczność modeli MAS w domenie modelowania mechanizmów rynkowych. Bez względu na sposób implementacji, wyniki symulacji przedstawiały charakterystyczną krzywą nasycania się rynku usługą telefonii komórkowej, którą przedstawia rysunek 1.



Rysunek1. Przebieg procesu nasycania się rynku usługą telefonii mobilnej w zależności od sposobu synchronizacji modelu – widoczne wyraźne podobieństwo kształtu krzywych życia produktu.

Źródło: Opracowanie własne.

W większości przetworzonych modeli nie stwierdzono znaczących zmian w precyzji działania programu w wyniku zwiększania liczby agentów z początkowej wielkości 1 000 – 2 500 do 10 000, a nawet 25 000. Z uwagi jednak na budowę systemów komputerowych stwierdzono efekt uboczny w postaci ponad proporcjonalnego wydłużenia się czasu wykonywania programu – zwiększenie liczby agentów do 10 000 powoduje zazwyczaj spadek wydajności o jeden lub dwa rzędy wielkości, a zwiększenie ich liczby do 25 000 nierzadko dziesięciokrotnie wydłuża czas niezbędny dla wykonania obliczeń związanych z przetworzeniem kolejnego IAP.

Tabela 2

Efektywność działania analizowanych modeli MAS wyznaczona w oparciu o łączny czas przetwarzania modelu oraz czas przetwarzania pojedynczego IAP tworzącego model

Liczba IAP	Metoda synchronizacji					
	synchroniczna		asynchroniczna		asynchroniczna pełna	
	łączny czas przetw. modelu (s)	czas przetw. pojedynczego IAP (s)	łączny czas przetw. modelu (s)	czas przetw. pojedynczego IAP (s)	łączny czas przetw. modelu (s)	czas przetw. pojedynczego IAP (s)
900	412	0,46	280	0,31	3 860	4,29
2 500	1 140	0,46	850	0,34	28 420	11,37
10 000	4 500	0,45	3 563	0,36	235 418	23,50

Źródło: Opracowanie własne.

Efekt wydłużenia czasu przetwarzania był szczególnie widoczny w przypadku modeli z aktywacją pełną asynchroniczną – stwierdzono kilkukrotne wydłużenie czasu przetwarzania pojedynczego agenta przy każdorazowym zwiększeniu liczby przetwarzanych IAP. Jest to o tyle zaskakujące, że sam algorytm ustalania losowej kolejności przetwarzania IAP działa bardzo sprawnie.

Kwestię wymagającą osobnego omówienia stanowi precyzja działania poszczególnych typów modeli w zależności od metody ich synchronizacji. Analizując różnice jakościowe pomiędzy różnymi typami modeli stwierdzono ich wysoką zbieżność, co potwierdza rysunek 1.

Z kolei zaobserwowano wyraźne różnice ilościowe w wynikach zwracanych przez poszczególne modele w zależności od sposobu aktywacji agentów, przy założeniu niezmiennych współczynników charakteryzujących model (por. rysunek 1). Aby zobrazować pojawiające się różnice przeprowadzono analizę statystyczną danych uzyskanych w oparciu o badane modele przetwarzania, obliczając: odchylenie standardowe oraz współczynniki determinacji i korelacji, przedstawione zbiorczo w tabeli 3.

Tabela 3

Różnice statystyczne w precyzji wyników zwracanych przez poszczególne modele

Współczynnik	Porównywane warianty synchronizacji modelu		
	synch.- asynch.	asynch.- asynch. pełna	synch.- asynch. pełna
odchylenie standardowe	104	85	62
współczynnik R^2 (determinacji)	0,71	0,92	0,60
współczynnik korelacji	0,84	0,96	0,78

Źródło: Opracowanie własne.

Obliczone statystyki wskazują na występowanie dużych różnic w wynikach zwracanych przez poszczególne modele, potwierdzając jednocześnie zbliżoną jakość rozwiązań. Zgodnie z tym, co dowodzą Huberman i Glance 1993 (por. [3]) – występują istotne różnice w wynikach w zależności od sposobu aktywacji agentów. Analiza literatury wykazuje powszechność stosowania asynchronicznej ich aktywacji, zarówno o stałym, jak i zmiennym kroku czasowym. Niezależnie od metody synchronizacji modelu oraz zakresu sąsiedztwa, obserwowano powstawanie charakterystycznych skupisk agentów o podobnych preferencjach, w wyniku wymiany informacji w ramach sieci powiązań personalnych. Efekt ten jest zgodny z obserwacjami rynków rzeczywistych i potwierdza funkcjonowanie wymiany informacji pomiędzy IAP. Analizując różnice w zależności od sposobu synchronizacji modelu oraz charakteru powiązań pomiędzy agentami (liczba „odpytywanych" sąsiadów oraz ich dobór), zaobserwowano znaczne rozbieżności zasięgu i mocy oddziaływań, wyraźnie widoczne na rysunku 2 jako skupiska IAP-klientów tej samej sieci. Skupiska te były mniejsze i mniej wyraźne w przypadku modeli o zmiennym kroku t (aktywacja asynchroniczna niepełna) oraz losowym doborze zakresu i liczebności sąsiedztwa. Zwiększenie liczby powiązań generalnie stabilizuje model poprzez utrwalanie istniejących zależności oraz zwiększenie zakresu informacji o otaczającym świecie dostępnych dla agenta, dzięki czemu wybory dokonywane przez niego są bliższe optymalnym (efekt losowości jest mniej widoczny). Wprowadzenie zmiennej liczby powiązań z otoczeniem przesuwa model w kierunku sieci o umiarkowanej liczbie połączeń. Zarówno w [5], jak i w [1] wprowadzenie parametru odpowiedzialnego za lojalność agentów spowodowało usztywnienie modelu i tworzenie większych struktur, odpowiednio: firm – w [5] i liczby klientów – w [1]. Trudno jednakże jednoznacznie określić, na ile obecność owych różnic

ma charakter artefaktów opisanych przez Axtella, a na ile jest efektem ścisłej współpracy pomiędzy agentami w sieci socjalnej.

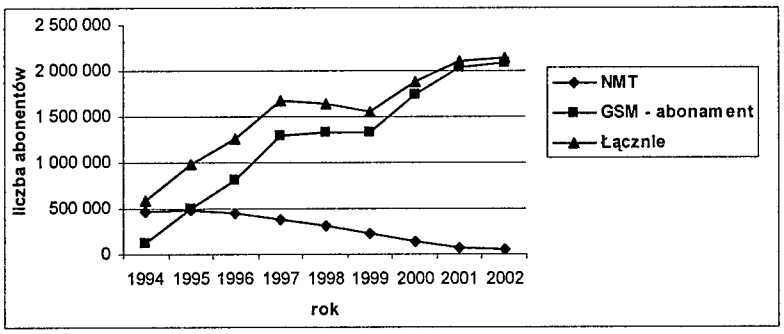


a) aktywacja synchroniczna b) aktywacja asynchroniczna c) aktywacja asynchr. pełna

Rysunek 2. Skupiska agentów o podobnych preferencjach w zależności od metody synchronizacji: kolorem białym oznaczono agenty nie związane z żadną siecią, odcieniami szarości – powiązane z jednym z trzech IAP-operatorów

Źródło: Opracowanie własne.

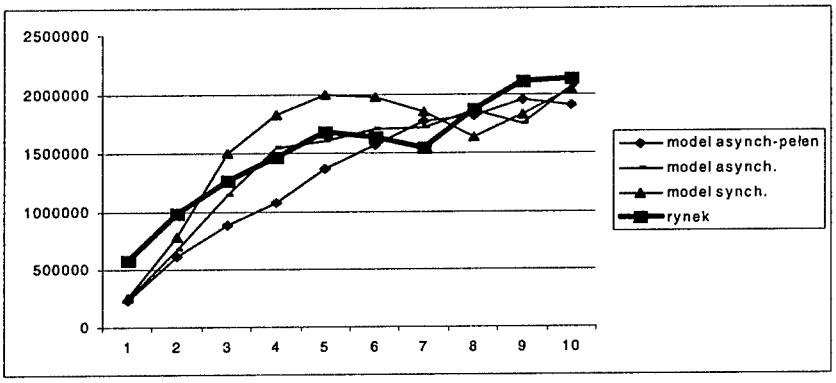
Kolejny krok prowadzonych analiz stanowiła ocena zdolności predykcyjnych przeanalizowanych powyżej modeli MAS. W tym celu porównano wyniki zwracane przez odpowiednio skalibrowany model z rynkiem rzeczywistym. Z uwagi na najdłuższy okres obecności usługi oraz dokładność dostępnych danych, do porównań wybrano rynek norweski. Warto zwrócić uwagę na gwałtowne przyspieszenie jego rozwoju obserwowane pod koniec lat 90. Odpowiedzialność za ten stan rzeczy ponoszą karty pre-paid wprowadzone w 1998 r. Osiągnęły one około 45% udział w rynku już w 1999 r. i stan ten utrzymuje się aż do dziś. Nie należy jednak uznawać ich za pełnoprawnego uczestnika rynku, gdyż operatorzy nie podają danych odnośnie faktycznego ich wykorzystania. Przykładowo w Norwegii stosowany jest 14 miesięczny okres oczekiwania przed dezaktywacją karty. Z tego też powodu w porównaniach uwzględniono tylko tych użytkowników, którzy opłacając regularnie abonament korzystają z usług bądź to telefonii mobilnej drugiej generacji (GSM), bądź pozostali przy rozwiązaniach analogowych (NMT). Szczegóły nasycania się rynku usługami świadczonymi w oparciu o abonament przedstawia rysunek 3.



Rysunek 3. Liczba użytkowników telefonii mobilnej GSM oraz NMT opłacających abonament w Norwegii w latach 1994-2002

Źródło: Opracowanie własne na podstawie [8].

W celu porównania danych, w badaniach ustalono okres trwania symulacji na 60 miesięcy oraz przemnożono uzyskane rezultaty modelu wyjściowego wykorzystującego 1200 IAP-klientów przez 4000, otrzymując w ten sposób wielkość porównywalną z liczbą mieszkańców Norwegii (4,5 mln). Rezultaty modelowania przedstawiono graficznie na rysunku 4 oraz opisano statystycznie w tabeli 4.



Rysunek 4. Wyniki przeprowadzonych symulacji z wykorzystaniem modeli o różnej konstrukcji, na tle rozwoju rynku telefonii mobilnej w Norwegii.

Tabela 4

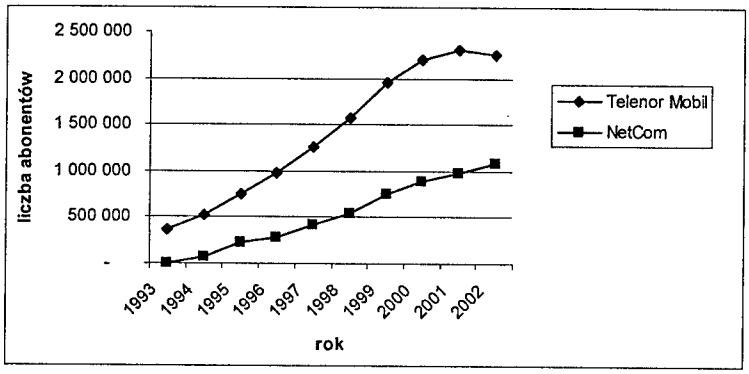
Różnice statystyczne w precyzji wyników zwracanych przez poszczególne modele

Współczynnik	Porównywane warianty symulacji względem rynku rzeczywistego		
	synchroniczny	asynchronizcny	w pełni asynchroniczny
odchylenie standardowe	313 tys.	226 tys.	313 tys.
współczynnik R^2 (determinacji)	0,79	0,90	0,91
współczynnik korelacji	0,89	0,95	0,95

Źródło: Opracowanie własne.

W wyniku przeprowadzonych analiz porównawczych stwierdzono brak podstaw do odrzucenia któregokolwiek z analizowanych modeli, gdyż wszystkie wykazują dużą zbieżność z danymi rzeczywistymi, mierzoną współczynnikami determinacji oraz korelacji.

Obserwowaną zbieżność danych rynkowych (por. rysunek 5) i wyników symulacji na poziomie poszczególnych operatorów, potwierdza efekt uzyskany w [9], a mianowicie tendencja do wyrównywania się rynkowych udziałów poszczególnych operatorów.



Rysunek 5. Udziały rynkowe norweskich operatorów Telenor Mobil oraz NetCom – widoczna utrata rynku przez Telenor na rzecz NetCom od 2000 r.

Na rynku norweskim obecni są dwaj operatorzy posiadający własną infrastrukturę sieciową – NetCom oraz Telenor Mobile, a także kilku tzw. operatorów wirtualnych (zakupujących pojemności od wspomnianych dwóch i sprzedających je pod własną marką, jednak o marginalnych udziałach rynkowych). Na przestrzeni kilku ostatnich lat można zaobserwować zdobywanie rynku przez NetCom kosztem Telenor i odrabianie początkowych strat związanych z późniejszym jego startem na rynku. Obserwowana tendencja jest zbieżna z tym, co się dzieje na rynku telefonii mobilnej w Polsce i w innych krajach europejskich.

6. ZAKOŃCZENIE

W wyniku przeprowadzonych eksperymentów potwierdzono znaczny wpływ budowy modelu na jego działanie. Wykazano, że sposób aktywacji (przetwarzania) agentów decyduje o wynikach zwracanych przez model, zaś pozostałe parametry jego budowy mają znaczenie drugorzędne – w ramach eksperymentów realizowanych z określonym typem synchronizacji modelu nie stwierdzono wyraźnych różnic w sposobie kształtowania się rynku. Ponadto stwierdzono, że zwiększanie liczby agentów ponad określony poziom nie jest celowe, gdyż nie tylko nie wnosi nowej jakości do pozyskiwanych w wyniku procesu modelowania danych, ale niepotrzebnie, często nawet ponad proporcjonalnie, wydłuża czas trwania symulacji.

Zgromadzony materiał empiryczny nie pozwala jeszcze wskazać optymalnej budowy modelu i metody jego przetwarzania. Trudno również jednoznacznie stwierdzić, jaki charakter mają skupiska IAP-klientów. Dostępne dane nie pozwalają precyzyjnie określić, jak rozległe są zależności pomiędzy ludźmi powiązanymi w sieci socjalne i na ile są one silne.

Reasumując, można uznać zdolność stworzonej rodziny modeli do opisywania w dostatecznym stopniu zjawisk zachodzących na rynku telefonii mobilnej. Uzyskane rezultaty pozwalają wnioskować, że obecnie sektor ten będzie dążył do stabilizacji na określonym poziomie nasycenia przy umiarkowanych jego wahaniach. Sytuacja ulegnie zmianie najprawdopodobniej w momencie wprowadzenia usługi alternatywnej, która spowoduje odpływ klientów w jej kierunku.

BIBLIOGRAFIA

[1] Sworowski R., Lasota H.: *Systemy wieloagentowe w prognozowaniu inwestycyjnym w telekomunikacji*, Krajowe Sympozjum Telekomunikacji 2003, Bydgoszcz.
[2] Gomółka Z.: *Elementy ogólnej teorii systemów*, Uniwersytet Szczeciński, Rozprawy i studia T(CCXXXIII) 159, Szczecin 1994.
[3] Epstein, J.M., Axtell R.: *Growing Artificial Societies: Social Science from the Bottom Up*, Brookings Press and MIT Press: Washington, D.C., and Cambridge, Mass.
[4] Chen, S.-H., Yeh C.-H.: *Modeling Speculators with Genetic Programming*. W: Angeline et al. 1997.
[5] Axtell R.: *Why agents? On the varied motivations for agent computing in the social sciences*, Working Paper No. 17, Nov. 2000, The Brookings Institution, Washington.
[6] Arthur W.B., Holland J.H., LeBaron B., Palmer R., Tayler P.: *Asset Pricing under Endogenous Expectations in an Artificial Stock Market*, Working paper 1996.
[7] Strzała K.: *Zastosowanie uogólnionych metod sterowania optymalnego do podejmowania decyzji gospodarczych*, Wydawnictwo UG, Sopot 1994.
[8] The Norwegian Post and Telecommunications Authority: *The Norwegian Telecom Market 2002*, report 2003.
[9] Sworowski R., Lasota H.: *Systemy wieloagentowe w modelowaniu rynku telekomunikacyjnego*, I KKTI, Politechnika Gdańska, Gdańsk 2003.

A STUDY OF THE INFLUENCE OF MULTIAGENT MODEL STRUCTURE ON ITS PREDICTION POWER

Summary

The article examines how different parameters of the multiagent model (MAS) influence its efficiency and efficacy in the IT market modeling. The growth process is exemplified by mobile telephony market with three competing operators. A demand model has been applied in MAS simulator and three different agent activating mechanisms have been tested, leading to different types of dynamic market equilibriums. The prediction power of the model has been estimated, the results being compared with data from a chosen real market, as well as its computing efficiency, depending on a number of agents and their activation schemes.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Zbigniew Buchalski

Instytut Cybernetyki Technicznej, Politechnika Wrocławska

ALOKACJA ZADAŃ W SYSTEMIE RÓWNOLEGLE PRACUJĄCYCH MASZYN

Streszczenie

Praca dotyczy zagadnienia czasowo-optymalnego szeregowania n zadań niezależnych niepodzielnych i rozdziału zasobów do m różnych maszyn równoległych. Zakłada się, że występuje stałość przydziału zasobów w czasie wykonywania całego zbioru zadań. Dla zadanej funkcji czasu realizacji zadań sformułowano model matematyczny zagadnienia i podano algorytm heurystyczny. Przedstawiono wyniki eksperymentów obliczeniowych.

1. WPROWADZENIE

Równoległe systemy przetwarzania informacji należą do jednych z najczęściej spotykanych systemów w praktyce. Ich intensywny rozwój pociągnął za sobą wzrost zainteresowania problematyką szeregowania zadań w systemach maszyn pracujących równolegle (niezależnie). Szczególnego znaczenia nabiera problem minimalizacji długości uszeregowania zadań na maszynach [3, 8, 9, 14].

Od wielu lat prowadzi się intensywne badania problematyki czasowo-optymalnego szeregowania zadań i rozdziału zasobów [2, 4, 5, 6, 7, 10, 11, 12, 13]. Zadania optymalizacji zarówno dyskretnej, jak i ciągłej należą do klasy problemów bardzo trudnych zarówno z teoretycznego, jak i obliczeniowego punktu widzenia i najczęściej należą do klasy problemów NP-zupełnych.

Czasy wykonywania zadań mogą być z góry zadane lub też mogą zmieniać się w zależności od współdzielonych zasobów. Zadania mogą być pewnymi procesami dynamicznymi opisywanymi za pomocą skomplikowanych modeli matematycznych. Pomiędzy zadaniami mogą występować pewne ograniczenia kolejnościowe lub mogą być im przypisane pewne wagi (priorytety). Mogą też być zadane terminy dostępności lub pożądane terminy zakończenia wykonywania poszczególnych zadań. Wykonywanie określonych zbiorów zadań jest zwykle oceniane z punktu widzenia różnych kryteriów rozpatrywanych pojedynczo bądź też wspólnie (podejście wielokryterialne).

Z formalnego punktu widzenia omawiane problemy można w sposób ogólny scharakteryzować następująco: dany jest zbiór zadań, zbiór maszyn oraz ograniczone zasoby przeznaczone do wykonywania tych zadań. Należy znaleźć takie dopuszczalne uszeregowanie tych zadań na maszynach oraz taki rozdział ograniczonych zasobów pomiędzy

zadania lub maszyny, na których te zadania są wykonywane, aby przyjęte kryterium efektywności osiągnęło optimum przy zadanych ograniczeniach.

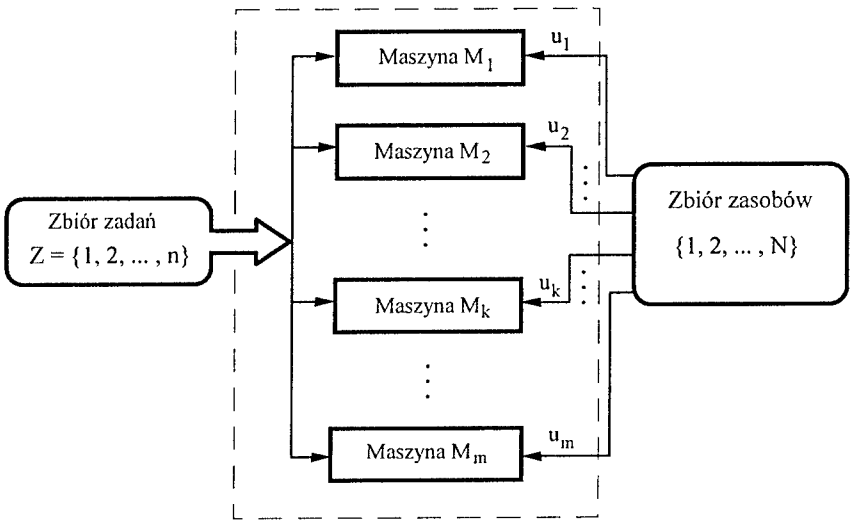
Doświadczenia z podejmowania prób optymalizacji tego typu problemów wskazują na możliwości osiągnięcia tą drogą znacznych efektów ekonomicznych, co w przypadku np. systemów wieloprocesorowych przy stale rosnącym zapotrzebowaniu na moc obliczeniową nie pozostaje bez znaczenia. Z tego punktu widzenia istnieje potrzeba prowadzenia badań w tym zakresie i to zarówno podstawowych, jak i stosowanych.

Problemy szeregowania zadań i rozdziału zasobów należą jednakże do bardzo trudnych zagadnień optymalizacyjnych. Pojawiają się bardzo trudne zagadnienia rozstrzygnięcia złożoności obliczeniowej rozpatrywanych problemów. Wyniki teorii złożoności obliczeniowej oraz rozmiar problemów praktycznych w sposób jednoznaczny eliminują z rozważań algorytmy dokładne, pozostawiając do zastosowania praktycznego jedynie algorytmy heurystyczne umożliwiające rozwiązanie postawionych problemów w krótkim czasie z zadowalającą dokładnością [1].

W prezentowanej pracy przedstawiono pewien algorytm heurystyczny wyznaczający czasowo-optymalne uszeregowanie n zadań na m maszynach równoległych. Jest to algorytm wielomianowy, a więc jest algorytmem efektywnym. Złożoność obliczeniowa tego typu algorytmów jest oszacowana w licznych opracowaniach monograficznych, np. w [1]. Przeprowadzono badania numeryczne zaproponowanego w pracy algorytmu dla losowo wybranych zestawów danych.

2. SFORMUŁOWANIE PROBLEMU, MODEL MATEMATYCZNY

Rozpatrzmy system maszyn równoległych przedstawiony na poniższym rysunku:



Rys. 1. System maszyn równoległych

Na system maszyn równoległych nakładamy następujące założenia:

(i) posiada m różnych maszyn $M = \{1, 2, \ldots, k, \ldots, m\}$, na których należy wykonać n niezależnych zadań $Z = \{1, 2, \ldots, i, \ldots, n\}$,

(ii) zadanie może być wykonywane na dowolnej maszynie i w trakcie jego wykonywania nie może być przerywane,

(iii) liczba zadań do wykonania jest większa od liczby maszyn $n > m$,

(iv) realizacja każdego z zadań na maszynach musi następować niezwłocznie po zakończeniu wykonywania poprzedniego zadania lub nastąpić w chwili zerowej, gdy zadanie realizowane jest jako pierwsze na jednej z maszyn.

Niech N oznacza globalną ilość zasobów nieodnawialnych, a przez u_k oznaczmy tą część zasobów, które zostaną przydzielone k-tej maszynie w trakcie wykonywania zadań uszeregowanych na tej maszynie. Ograniczenie dotyczące zasobów jest następujące:

$$\sum_{k=1}^{m} u_k \le N, \quad u_k \ge 0, \quad 1 \le k \le m.$$

Czas wykonywania i-tego zadania na k-tej maszynie określony jest przez funkcję $T_i(u_k, k)$:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in \{1,2,...,N\}, \quad 1 \le k \le m, \quad 1 \le i \le n. \tag{2.1}$$

Parametry $a_{ik} > 0$ i $b_{ik} > 0$ charakteryzują i-te zadanie i k-tą maszynę.

Należy znaleźć takie uszeregowanie zadań na maszynach i taki przydział ograniczonych zasobów do maszyn równoległych, aby minimalizować czas wykonania całego zbioru zadań T_{zak}.

Jeżeli oznaczymy przez $Z_k \subset Z$ zbiór zadań uszeregowanych na k-tej maszynie, to T_{zak} znajdziemy rozwiązując następujący problem minimalizacyjny:

$$T_{zak} = \min_{\substack{Z_1,Z_2,...,Z_m \\ u_1,u_2,...,u_m}} \max_{1 \le k \le m} \left\{ \sum_{i \in Z_k} T_i(u_k, k) \right\} \tag{2.2}$$

Ograniczenia nałożone na rozwiązanie tego problemu są następujące:

(i) $Z_r \cap Z_s = \phi;\ r,s = 1,2,...,m,\ r \ne s, \bigcup_{k=1}^{m} Z_k = Z$

(ii) $\sum_{k=1}^{m} u_k \le N,$

(iii) $u_1, u_2, ..., u_m$ – całkowite dodatnie.

Dla uproszczenia problemu przyjmiemy najpierw, że zasoby nieodnawialne $u_1, u_2, ..., u_m$ są typu ciągłego. Przy tym założeniu wyznaczymy rozwiązanie optymalne, a następnie zaokrąglimy otrzymane wartości zasobów do najbliższych liczb naturalnych. Tak więc, czas T_{zak} znajdziemy rozwiązując następujący problem minimalizacji dyskretno-ciągłej:

$$T_{zak} = \min_{\substack{Z_1,Z_2,...,Z_m \\ u_1,u_2,...,u_m}} \max_{1 \le k \le m} \left\{ \sum_{i \in Z_k} T_i'(u_k, k) \right\} \tag{2.3}$$

przy następujących ograniczeniach:

(i) $Z_r \cap Z_s = \phi;\ r,s = 1,2,...,m,\ r \ne s, \bigcup_{k=1}^{m} Z_k = Z$

(ii) $\sum_{k=1}^{m} u_k \le N, \quad u_k \ge 0, \quad k = 1,2,...,m,$

gdzie: T_i': $[0,N]\times\{1,2,..,m\} \to R^+$ jest rozszerzeniem następującej funkcji

$T_i : \{1,2,...,N\}\times\{1,2,..,m\} \to R^+$ i określone jest przez funkcję:

$$T_i'(u_k,k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in [0,N], \quad 1 \le k \le m, \quad 1 \le i \le n. \tag{2.4}$$

Do rozwiązania postawionego problemu pomocny będzie następujący lemat:

LEMAT 1

Jeżeli $u_k^*, Z_k^*, k = 1,2,...,m$ są rozwiązaniami zadania (2.3), to:

(i) $\sum_{k=1}^{m} u_k^* = N; \quad u_k^* > 0, \quad k : Z_k^* \ne \phi, \quad k = 1,2,...,m;$

$$u_k^* = 0, \quad k : Z_k^* = \phi, \quad k = 1,2,...,m;$$

(ii) $\sum_{i \in Z_k^*} T_i'(u_k^*,k) = const, \quad k : Z_k^* \ne \phi, \quad k = 1,2,...,m.$

Warunek (i) w **LEMACIE 1** oznacza, że w przydziale czasowo-optymalnym zasobów i zadań do maszyn wykorzystuje się wszystkie jednostki zasobów, a warunek (ii), że czasy pracy tych maszyn, na których wykonywane są jakieś zadania, są identyczne.

Zdefiniujmy funkcję $F(Z_1, Z_2,..., Z_m)$ określoną dla m zbiorów $Z_1, Z_2,..., Z_m$, dla których zachodzi ograniczenie (i) dla wzoru (2.3). Wartość tej funkcji jest rozwiązaniem następującego układu równań:

$$\begin{cases} \sum_{i \in Z_k} a_{ik} + \dfrac{\sum_{i \in Z_k} b_{ik}}{u_k} = F(Z_1, Z_2,..., Z_m); \quad k : Z_k \ne \varnothing, \quad k = 1,2,...,m \\ \sum_{k=1}^{m} u_k = N; \quad u_k > 0, \quad k : Z_k \ne \varnothing, \quad k = 1,2,...,m. \end{cases} \tag{2.5}$$

Wykorzystując **LEMAT 1** oraz (2.5) zadanie minimalizacji (2.3) można przedstawić w następującej postaci:

$$T_{zak} = \min_{Z_1, Z_2,..., Z_m} F(Z_1, Z_2,..., Z_m), \tag{2.6}$$

przy ograniczeniach:

(i) $Z_r \cap Z_s = \phi, \quad r,s = 1,2,...,m, \quad r \ne s,$

(ii) $\bigcup_{k=1}^{m} Z_k = Z.$

Jeżeli $Z_1^*, Z_2^*,.., Z_m^*$ jest rozwiązaniem zadania (2.6), to $u_k^*, Z_k^*, k = 1,2,...,m,$ gdzie

$$u_k^* = \begin{cases} \dfrac{\sum_{i \in Z_k^*} b_{ik}}{F(Z_1^*, Z_2^*,..., Z_m^*) - \sum_{i \in Z_k^*} a_{ik}}; & k : Z_k^* \ne \phi, \quad 1 \le k \le m, \\ 0 \qquad\qquad\qquad\qquad\quad ; & k : Z_k^* = \phi, \quad 1 \le k \le m \end{cases}$$

jest rozwiązaniem zadania (2.3).

3. ALGORYTM HEURYSTYCZNY

W związku z tym, że zasoby przydzielamy do maszyn, maszyny te różnią się pod względem szybkości wykonywania zadań. Im więcej zasobów u_k przydzielimy k-tej maszynie, tym będzie ona szybsza.

Formuła przydziału zasobów do maszyn jest następująca:

- zakładamy, że maszyną najszybszą jest maszyna m-ta, a maszyną najwolniejszą jest maszyna pierwsza,
- przyjmujemy istnienie tzw. współczynnika podziału zasobów $\alpha; 0 < \alpha < 1$ różnicującego maszyny pod względem szybkości realizacji zadań.

Jeżeli maszynie najszybszej przydzielimy u_m jednostek zasobów, wg zależności:

$$u_m = \frac{N}{1 + \sum_{k=1}^{m-1} [(m-k)\cdot\alpha]}, \tag{3.1}$$

to pozostałe maszyny otrzymają liczbę zasobów określoną następującą zależnością:

$$u_k = (m-k)\cdot\alpha\cdot u_m, \quad k = 1, 2, \ldots, m-1. \tag{3.2}$$

Poszczególne kroki algorytmu heurystycznego są następujące:

Krok 1. Oblicz czasy wykonywania zadań na poszczególnych maszynach dla $u_k = \frac{N}{m}$ i dla losowo generowanych parametrów a_{ik}, b_{ik} wg wzoru $T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}$, $i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, m$.

Krok 2. Uszereguj malejąco zadania wg czasu ich trwania, tworząc listę L tych zadań.

Krok 3. Oblicz średni czas $T_{śr}$ wykonywania zadań przez każdą z maszyn wg wzoru

$$T_{śr} = \frac{\sum_{i=1}^{n} T_i(u_k, k)}{m}; \quad i = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, m.$$

Krok 4. Przydzielaj kolejne zadania z listy L do pierwszej wolnej maszyny aż do momentu, gdy suma czasów wykonywania tych zadań nie przekroczy $T_{śr}$ i usuń te zadania z listy L.

Krok 5. Przydziel kolejne zadania z listy L do następnych wolnych maszyn wg procedury z **kroku 4** aż do wyczerpania zbioru maszyn.

Krok 6. Nieprzydzielone w **krokach 4** i **5** zadania z listy L przydziel do maszyn wg algorytmu LPT (*Longest Processing Time*) aż do momentu wyczerpania się listy zadań.

Krok 7. Dla zadanego współczynnika podziału zasobów α przydziel zasoby u_k, wyliczone ze wzorów (3.1) i (3.2).

Krok 8. Oblicz czas zakończenia wykonywania wszystkich zadań T_{zak} dla uszeregowania zadań na maszynach utworzonego w **krokach 4÷6** i dla liczby zasobów u_k $k = 1, 2, \ldots, m$ przydzielonych maszynom w **kroku 7**.

Krok 9. Powtórz **kroki 7 i 8** dla kilku różnych wartości współczynnika α.

Krok 10. Porównaj wartości czasów zakończenia wykonywania wszystkich zadań T_{zak} z kolejnych prób. Wybierz najkrótszy z czasów T_{zak}.

Krok 11. Wyznacz dyskretne ilości zasobów $\hat{u}_k, k = 1, 2, ..., m$ według zależności:

$$\hat{u}_{\alpha(k)} = \begin{cases} \lfloor u_{\alpha(k)} \rfloor + 1; & k = 1, 2, ..., \Delta, \\ \lfloor u_{\alpha(k)} \rfloor & k = \Delta + 1, \Delta + 2, ..., m, \end{cases}$$

gdzie $\Delta = N - \sum_{j=1}^{m} \lfloor u_j \rfloor$ oraz α jest permutacją elementów zbioru $M=\{1,2,..., m\}$ taką, że $u_{\alpha(1)} - \lfloor u_{\alpha(1)} \rfloor \geq u_{\alpha(2)} - \lfloor u_{\alpha(2)} \rfloor \geq ... \geq u_{\alpha(m)} - \lfloor u_{\alpha(m)} \rfloor$.

Jeżeli istnieją takie maszyny, którym przydzielono zerowe ilości zasobów, to przydziel każdej z tych maszyn po jednej jednostce zasobu pobierając je z kolejnych maszyn poczynając od maszyny, której przydzielono największą ilość zasobów.

Przeprowadzono eksperymenty obliczeniowe na bazie przedstawionego algorytmu dla dziewięciu zwiększających się kolejno wartości współczynnika podziału zasobów α z przedziału [0.1, 0.2, ... , 0.9] ze skokiem co 0.1. Parametry charakteryzujące i-te zadanie i k-tą maszynę a_{ik} , b_{ik} wylosowane zostały ze zbioru {5.0, 10.0, ... , 50.0} przez generator o jednostajnym rozkładzie prawdopodobieństwa. Dla każdej kombinacji n i m wygenerowano 50 instancji. Rezultaty analizy porównawczej algorytmu heurystycznego skonstruowanego dla potrzeb niniejszej pracy i znanego z literatury algorytm LPT przedstawione zostały w tablicy 3.1.

Tablica 3.1

Wyniki analizy porównawczej algorytmu heurystycznego i algorytmu LPT

n/m	Liczba instancji, dla których:			Δ^H	S^H	S^{LPT}
	$T_{zak}^H < T_{zak}^{LPT}$	$T_{zak}^H = T_{zak}^{LPT}$	$T_{zak}^H > T_{zak}^{LPT}$	%	sek	sek
20/2	28	1	21	1,5	1,2	1,3
20/4	32	2	16	3,1	1,6	1,7
20/6	36	3	11	3,8	2,8	2,7
20/8	37	1	12	2,9	2,9	3,1
40/2	30	2	18	2,4	2,2	2,4
40/4	31	3	16	3,6	2,7	2,6
40/6	29	4	17	3,5	3,1	3,2
40/8	35	0	15	3,9	3,6	3,7
60/2	30	1	19	2,9	3,4	3,2
60/4	29	3	18	1,9	4,0	3,9
60/6	32	0	18	2,8	4,2	4,4
60/8	31	2	17	2,6	4,9	4,8
80/2	29	3	18	2,4	5,1	5,2
80/4	30	4	16	3,2	5,4	5,3
80/6	31	2	17	3,8	7,2	7,3
80/8	34	1	15	4,1	8,2	8,4

W tablicy 3.1 występują następujące wielkości:

n – liczba zadań,

m – liczba maszyn,

T_{zak}^{H} – czas zakończenia wykonywania wszystkich zadań ze zbioru Z przy wykorzystaniu algorytmu heurystycznego,

T_{zak}^{LPT} – czas zakończenia wykonywania wszystkich zadań ze zbioru Z przy wykorzystaniu algorytmu LPT,

Δ^{H} – średnia procentowa poprawa czasu T_{zak}^{H} w stosunku do T_{zak}^{LPT} :

$$\Delta^{H} = \frac{T_{zak}^{LPT} - T_{zak}^{H}}{T_{zak}^{H}} \cdot 100\%$$

S^{H} – średni czas obliczeń dla algorytmu heurystycznego,

S^{LPT} – średni czas obliczeń dla algorytmu LPT,

4. UWAGI KOŃCOWE

Przedstawione w poprzednim rozdziale eksperymenty obliczeniowe pokazują, że jakość szeregowania zadań na równoległych maszynach na bazie zaproponowanego w pracy algorytmu heurystycznego uległa poprawie w stosunku do szeregowania za pomocą znanego z literatury algorytmu LPT. Kilkuprocentowa poprawa czasu T_{zak}^{H} w stosunku do T_{zak}^{LPT} może być zachętą do dalszych prac nad efektywnymi algorytmami heurystycznymi.

Zastosowanie opracowanego algorytmu jest wskazane przede wszystkim dla systemów o dużej liczbie zadań, gdyż wówczas średnia procentowa poprawa Δ^{H} jest największa. Zaproponowana w pracy heurystyka może służyć zarówno do szeregowania programów w systemach procesorów równoległych, jak i rozdziału operacji na równoległe maszyny w systemach produkcyjnych.

BIBLIOGRAFIA

[1] Błażewicz J.: Złożoność obliczeniowa problemów kombinatorycznych, WNT, Warszawa 1988.

[2] Błażewicz J., Dell'Olmo P., Drozdowski M., Speranza M. G. : Scheduling multiprocessor tasks on three dedicated processors. Information Processing Letters 41, 1992, pp. 275-280.

[3] Błażewicz J., Drabowski M., Węglarz J.: Scheduling multiprocessor tasks to minimize schedule length. IEEE Transactions on Computers C-35, 1986, pp. 389-393.

[4] Boctor F. F.: A new and efficient heuristic for scheduling projects will resources restrictions and multiple execution models. European Journal of Operational Research, vol. 90, 1996, pp. 349-361.

[5] Buchalski Z.: Optimization of programs scheduling and primary memory allocation in multiprocessing computer systems. Information Systems Architecture and Technology ISAT'98, Wrocław 1998, pp. 246-253.

[6] Giaro K.: Wybrane zastosowania niestandardowych modeli kolorowania w szeregowaniu dwuprocesorowych zadań jednostkowych. Uczelniane Wydawnictwo Naukowo-Dydaktyczne, AGH, seria-Automatyka, półrocznik, tom 7, zeszyt 1-2, Kraków 2003, str. 105-111.

[7] Giaro K., Kubale M., Małafiejski M.: Szeregowanie zadań jednostkowych w systemie otwartym bez obustronnych przestojów. Zeszyty Naukowe Pol. Śl., seria-Automatyka, 117, Gliwice 1996, str. 29-36.

[8] Janiak A.: Single machine scheduling problem with a common deadline and resource dependent release dates. European Journal of Operational Research, vol. 53, 1991, pp. 317-325.
[9] Janiak A., Kovalyov M.: Single machine scheduling subject to deadlines and resources dependent processing times. European Journal of Operational Research, 1996, vol. 94, pp. 284-291.
[10] Józefowska J., Węglarz J.: On a methodology for discrete-continuous scheduling. European Journal of Operational Research, vol. 107, 1998, pp. 338-353.
[11] Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J.: Rozwiązywanie dyskretno-ciągłych problemów rozdziału zasobów przez dyskretyzację zasobu ciągłego. Zeszyty Naukowe Politechniki Śląskiej Nr 1474, seria – Automatyka, Gliwice 2000, z. 129, str. 221-229.
[12] Krawczyk H., Kubale M.: An approximation algorithm for diagnostic test scheduling in multicomputer systems. IEEE Trans. Comp., C-34, 1985, pp. 869-872.
[13] Kubale M., Giaro K.: Złożoność zwartego szeregowania zadań jednostkowych w systemie otwartym, przepływowym i mieszanym. Uczelniane Wydawnictwo Naukowo-Dydaktyczne AGH, seria-Automatyka, półrocznik, tom 5, zeszyt ½, Kraków 2001, str. 329-334.
[14] Nowicki E., Smutnicki C.: The flow shop with parallel machines. A Tabu search approach. European Journal of Operational Research 106, 1998, pp. 226-253.

TASKS ALLOCATION IN PARALLEL MULTIMACHINES SYSTEM

Summary

In the paper the problem of time-optimal scheduling of n independent nopreemtive tasks and resources allocation to m different parallel machines is considered. We assume, that is constancy of resources allocation in execution time all tasks set. For some tasks execution time function the mathematical model of this probelm is formulated and an heuristic algorithm is presented. Some results of executed numerical experiments are presented.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Dariusz Dereniowski

Katedra Algorytmów i Modelowania Systemów, Politechnika Gdańska

ANALIZA PRZYBLIŻONEGO ALGORYTMU DLA PROBLEMU SZUKANIA DRZEWA SPINAJĄCEGO O MINIMALNYM UPORZĄDKOWANYM INDEKSIE CHROMATYCZNYM

Streszczenie

Uporządkowane kolorowanie krawędzi grafu polega na takim etykietowaniu krawędzi liczbami naturalnymi, że każda ścieżka łącząca krawędzie o jednakowej barwie zawiera krawędź o kolorze wyższym. W pracy rozważamy problem kombinatoryczny MERST, polegający na znalezieniu, dla danego grafu G, jego drzewa spinającego, którego uporządkowany indeks chromatyczny jest najmniejszy spośród indeksów chromatycznych wszystkich drzew spinających. Makino, Uno i Ibaraki [1] podali przybliżony algorytm rozwiązujący powyższy problem w przypadku grafów ogólnych. W pracy podajemy lepszą funkcję dobroci od tej, która została podana w [1] oraz prezentujemy wyniki testów komputerowych zebranych podczas implementacji algorytmu. Testy obejmują zarówno porównanie funkcji dobroci, jak i oszacowań uporządkowanego indeksu chromatycznego drzew, na podstawie których uzyskano opisane funkcje dobroci.

1. WSTĘP

Niech $G=(V,E)$ będzie grafem prostym, gdzie V jest zbiorem wierzchołków ($|V|=n$), natomiast E jest zbiorem krawędzi ($|E|=m$). Funkcja $c:E(G)\rightarrow\{1,...,k\}$ jest uporządkowanym k-pokolorowaniem krawędziowym grafu G, gdy każda ścieżka łącząca krawędzie e_1,e_2 takie, że $c(e_1)=c(e_2)$ zawiera krawędź e o własności $c(e)>c(e_1)$. Najmniejsza liczba k, dla której istnieje uporządkowane k-pokolorowanie, to uporządkowany indeks chromatyczny grafu, który oznaczamy symbolem $\chi_r'(G)$. Problem polegający na znalezieniu indeksu chromatycznego grafu jest NP-zupełny już w przypadku grafów dwudzielnych [2]. Problemowi kolorowania uporządkowanego drzew poświęcono dużo uwagi m.in. ze względu na jego zastosowania praktyczne [3]. Pierwszy znany algorytm był 2-przybliżony [4]. W [5] podano optymalny algorytm o złożoności $O(n^3\lg n)$. Obecnie wiadomo, iż problem można rozwiązać w czasie liniowym [6]. Problem szukania drzewa spinającego o minimalnym indeksie chromatycznym został zdefiniowany w pracy [1], gdzie podano dowód, iż problem ten jest w ogólności NP-trudny. W tym samym artykule opisano przybliżony algorytm o funkcji dobroci postaci

$$\frac{\min\{(\Delta^*-1)\lg n/\Delta^*, \Delta^*-1\}}{\lg(\Delta^*+1)-1}, \tag{1.1}$$

gdzie Δ^* jest maksymalnym stopniem wierzchołka drzewa spinającego, którego maksymalny stopień wierzchołka jest najmniejszy spośród wszystkich drzew spinających. W powyższym wyrażeniu oraz w całej pracy lg oznacza logarytm o podstawie 2. Problem był również rozważany w pracy [7], gdzie wskazano jego potencjalne zastosowania. Jednym z nich jest planowanie procesu równoległego złączania tabel w relacyjnych bazach danych. Przy pewnych założeniach, graf opisujący sposób łączenia relacji jest grafem progowym i takie grafy były rozważane w [7], gdzie podano liniowy algorytm znajdujący ich drzewa spinające o minimalnym uporządkowanym indeksie chromatycznym.

2. ALGORYTM

W poniższym rozdziale zajmiemy się analizą funkcji dobroci przybliżonego algorytmu, znajdującego i kolorującego drzewo spinające o niskim indeksie chromatycznym dla danego grafu prostego G. Zanim opiszemy ów algorytm, podamy oszacowania indeksu chromatycznego drzew, które będą potrzebne w dalszej części.

Twierdzenie 1 ([8]) *Istnieje* 1*-absolutnie przybliżony algorytm dla problemu szukania drzewa spinającego o minimalnym stopniu.* □

Algorytm dla problemu MERST (ang. *minimum edge ranking spanning tree*), podany w pracy [1], składa się z dwóch etapów. Pierwszy polega na wykorzystaniu powyższego twierdzenia, w celu znalezienia drzewa spinającego. Drugi etap, to uporządkowane pokolorowanie wcześniej znalezionego drzewa. Taki algorytm jest przybliżony, gdyż pierwszy z etapów jest suboptymalny. Ponadto, znalezienie drzewa spinającego o minimalnym stopniu nie oznacza znalezienia tego o minimalnym indeksie chromatycznym.

Poniżej prezentujemy dwa oszacowania, dolne oraz górne, na uporządkowany indeks chromatyczny drzew. Symbol Δ_T będzie w dalszej części pracy używany do oznaczania stopnia drzewa T.

Lemat 1 ([1]) *Dla dowolnego drzewa T zachodzi* $\chi_r'(T) \geq \max\{\Delta_T, \lceil \lg n \rceil\}$. □

Lemat 2 ([1]) *Jeśli T jest drzewem, to zachodzi oszacowanie górne*

$$\chi_r'(T) = \lceil \lg n \rceil, \quad \text{gdy } \Delta_T = 1,2$$
$$\chi_r'(T) \leq \frac{(\Delta_T - 2)\lg n}{\lg \Delta_T - 1}, \quad \text{gdy } \Delta_T > 2.$$

Przez Δ^* oznaczmy stopień drzewa spinającego, dla którego ten parametr jest minimalny oraz niech T_o,T będą odpowiednio drzewem spinającym o minimalnym indeksie chromatycznym oraz drzewem znalezionym przez algorytm z twierdzenia 1. Wówczas, na podstawie lematu 1 zastosowanego do T_o oraz lematu 2, zastosowanego do T można napisać [1]:

$$\frac{\chi_r'(T)}{\chi_r'(T_o)} \leq \frac{(\Delta^*-1)\lg n/(\lg(\Delta^*+1)-1)}{\max\{\Delta^*,\lceil \lg n \rceil\}} \leq \frac{\min\{(\Delta^*-1)\lg n/\Delta^*, \Delta^*-1)\}}{\lg(\Delta^*+1)-1} \tag{2.1}$$

Powyższe wyrażenie stanowi wzór na funkcję dobroci (która będzie dalej oznaczana symbolem $R_1(\Delta^*,m)$) algorytmu, który dla podanego na wejściu grafu prostego G oblicza, korzystając z algorytmu z twierdzenia 1, drzewo spinające o niskim stopniu, a następnie koloruje jego krawędzie za pomocą liczby kolorów równej co najwyżej wartości górnego oszacowania z lematu 2.

Poniżej przytaczamy inne oszacowanie górne indeksu chromatycznego drzew, które następnie zostanie wykorzystane do wyprowadzenia nowej funkcji dobroci omawianego algorytmu.

Lemat 3 ([2]) *Dla dowolnego spójnego drzewa T o $n > 2$ wierzchołkach zachodzi oszacowanie górne*

$$\chi_r'(T) \leq \Delta_T \log_{\Delta_T} m,$$

gdzie Δ_T jest maksymalnym stopniem wierzchołka w T. □

Poniższy lemat porównuje dwa wymienione wcześniej oszacowania. Zostanie on wykorzystany przy opisie własności nowo skonstruowanej funkcji dobroci.

Twierdzenie 2 *Jeżeli $4 \leq \Delta < n$, to zachodzi nierówność*:

$$\Delta \log_\Delta m < \frac{(\Delta-2)\lg n}{\lg\Delta - 1}.$$

Dowód: Określamy funkcję pomocniczą:

$$f(\Delta) = \Delta - 2\lg\Delta.$$

Funkcja f jest rosnąca oraz $f(4) = 0$. Stąd $f(\Delta) \geq 0$, dla $\Delta \geq 4$. Oznacza to, że dla $\Delta \geq 4$ prawdziwa jest nierówność

$$2\lg\Delta \leq \Delta.$$

Stąd otrzymujemy

$$\Delta\lg\Delta - \Delta \leq \Delta\lg\Delta - 2\lg\Delta,$$

$$\frac{\Delta}{\lg\Delta} \leq \frac{\Delta-2}{\lg\Delta-1}.$$

Mnożąc obustronnie nierówności przez $\lg n$ otrzymujemy

$$\Delta\frac{\lg n}{\lg\Delta} \leq (\Delta-2)\frac{\lg n}{\lg\Delta-1}.$$

co kończy dowód, gdyż $\log_\Delta n = \lg n/\lg\Delta$ oraz $\log_\Delta m < \log_\Delta n$. □

Na podstawie lematów 1 oraz 3 można napisać

$$\frac{\chi_r'(T)}{\chi_r'(T_o)} \leq \frac{(\Delta^*+1)\log_{\Delta^*+1}(n-1)}{\max\{\Delta^*, \lceil \lg n \rceil\}}, \tag{2.2}$$

gdzie, jak poprzednio, T jest drzewem będącym wynikiem działania algorytmu z lematu 1, natomiast T_o jest drzewem spinającym o minimalnym indeksie chromatycznym. Powyższa funkcja dobroci będzie oznaczana przez $R_2(\Delta^*,n)$.

Twierdzenie 3 *Jeżeli* $\Delta^* \geq 3$, *to* $R_2(\Delta^*,n) < R_1(\Delta^*,n)$.

Dowód: Z lematu 2 oraz założenia $\Delta^* \geq 3$ wynika, że

$$(\Delta^*+1)\log_{\Delta^*+1} m < \frac{(\Delta^*-1)\lg n}{\lg(\Delta^*+1)-1}.$$

Mnożąc nierówność obustronnie przez $1/\max\{\Delta^*, \lceil \lg n \rceil\}$ otrzymujemy

$$R_2(\Delta^*,n) < \frac{(\Delta^*-1)\lg n}{\lg(\Delta^*+1)-1} \cdot \frac{1}{\max\{\Delta^*, \lceil \lg n \rceil\}} \leq R_1(\Delta^*,n),$$

co kończy dowód. □

Trudność przy bezpośrednim wykorzystaniu wzorów $R_1(\Delta^*,n)$ i $R_2(\Delta^*,n)$ polega na tym, że parametr Δ^* nie jest znany. Dysponujemy jedynie wartością Δ_T, gdzie T jest drzewem spinającym znalezionym w pierwszej fazie algorytmu. Możemy pominąć przypadki, gdy $\Delta^* = 0,1$, gdyż jeśli G jest spójny, to są to sytuacje trywialne. Założenie spójności grafu nie zmniejsza ogólności rozważań, ponieważ można zastosować opisany algorytm do każdej składowej spójności grafu osobno. Wówczas wyznaczenie funkcji dobroci dla poszczególnych składowych pozwala oszacować błąd rozwiązania dla całego grafu G. Przez $R_i'(\Delta_T,n)$ oznaczmy funkcję dobroci, skonstruowaną na podstawie $R_i(\Delta^*,n)$, $i = 1,2$ następująco:

$$R_i'(\Delta_T,n) = \max\{R_i(\Delta_T - 1,n), R_i(\Delta_T,n)\}.$$

Zauważmy, że definicja jest poprawna, tzn. $R_i'(\Delta_T,n)$ jest funkcją dobroci opisanego algorytmu, co wynika z faktu, iż $\Delta^* \in \{\Delta_T - 1, \Delta_T\}$. Ponadto, obliczenie $R_i'(\Delta_T,n)$ nie wymaga znajomości parametru Δ^*, więc nowo skonstruowane funkcje są łatwe obliczeniowo.

W praktyce, do oszacowania błędu znalezionego rozwiązania można wykorzystać funkcję R_2. Warto jednak osobno rozważyć pewne przypadki szczególne, aby ostatecznie zdefiniować funkcję dobroci. Funkcję o tej własności oznaczmy przez $R(\Delta_T,n)$, którą zdefiniujemy rozpatrując trzy przypadki:

- *Przypadek* 1: $\Delta_T = 2$. Wówczas znalezione drzewo spinające jest ścieżką, co oznacza, że jest to optymalne rozwiązanie dla problemu MERST, więc możemy przyjąć $R(2,n) = 1$.
- *Przypadek* 2: $\Delta_T = 3$. W takiej sytuacji przyjmujemy

$$R(\Delta_T,n) = \min\{ R_1(\Delta_T,n), R_2(\Delta_T,n) \}.$$

Poprawność wynika z definicji funkcji R_i.

- *Przypadek* 3: $\Delta_T > 3$. Na mocy twierdzenia 3 oraz dyskusji powyżej możemy przyjąć: $R(\Delta_T,n) = R_2'(\Delta_T,n)$.

3. TESTY

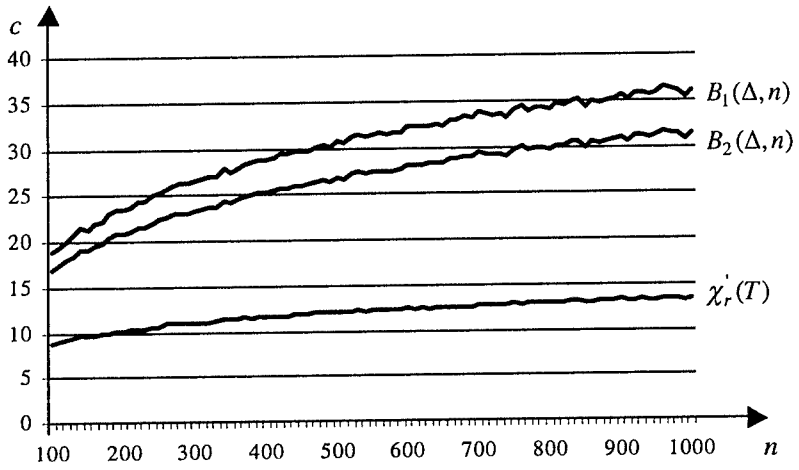
Poniżej prezentujemy wyniki doświadczeń komputerowych zebranych podczas implementacji algorytmu. Pierwszy z testów dotyczył porównania funkcji dobroci R_1 oraz R_2. Badano w jaki sposób kształtują się funkcje dobroci algorytmu w przypadku grafów losowych o różnych rozmiarach i gęstościach. Liczby wierzchołków generowanych grafów wynosiły $n = 100,200,...,1000$, natomiast ich gęstości $g = 0.005,0.01,0.05,0.1$. Małe wartości parametru g wynikają z faktu, iż do szukania drzewa spinającego został użyty algorytm 1-absolutnie przybliżony. Losowane grafy o większych gęstościach były zazwyczaj półhamiltonowskie, gdyż algorytm z twierdzenia 1 znajdował drzewa spinające będące ścieżkami, co prowadziło do optymalnego rozwiązania problemu MERST. W takich przypadkach badanie funkcji dobroci nie było uzasadnione. Grafy, które podawano na wejście algorytmu były spójne. Wyniki przedstawione w poniższej tabeli są uśrednione na podstawie 100 prób, przy czym każda próba kończyła się pokolorowaniem drzewa spinającego T o $\Delta_T \geq 3$ (przypadki, gdy $\Delta_T = 2$ były odrzucane). Dla ustalonych wartości n i g podano w tabeli kolejno: liczbę użytych kolorów, wartość funkcji $R_1(\Delta_T,n)$ oraz wartość $R_2(\Delta_T,n)$.

Porównanie funkcji $R_1(\Delta_T,n)$ oraz $R_2(\Delta_T,n)$

n	g = 0.005	g = 0.01	g = 0.05	g = 0.1
100	7.97 2.43 2.14	7.98 2.47 2.17	7.01 2.08 1.94	7.00 2.02 1.90
200	9.00 2.58 2.26	8.28 2.35 2.11	8.00 2.10 1.96	8.00 2.03 1.92
300	9.02 2.46 2.09	9.00 2.45 2.08	9.00 2.20 1.94	9.00 2.03 1.84
400	9.92 2.51 2.22	9.51 2.54 2.24	9.06 2.54 2.24	9.00 2.05 1.95
500	10.00 2.58 2.35	10.03 2.64 2.40	10.03 2.73 2.46	9.87 2.60 2.38
600	10.15 2.62 2.20	10.16 2.68 2.25	10.22 2.83 2.36	10.14 2.85 2.38
700	10.83 2.70 2.31	10.72 2.78 2.37	10.46 2.85 2.43	10.56 2.97 2.53
800	11.01 2.71 2.37	11.14 2.89 2.50	11.44 2.95 2.55	10.86 2.94 2.55
900	11.05 2.75 2.44	11.06 2.80 2.47	11.53 2.96 2.59	11.58 3.06 2.67
1000	11.31 2.79 2.51	11.20 2.85 2.55	11.55 3.04 2.68	11.98 3.01 2.65

Druga część testów dotyczyła oszacowań uporządkowanego indeksu chromatycznego drzew. Ze wzorów (2.1) oraz (2.2), definiujących badane wcześniej funkcje dobroci wynika, że funkcje R_1 i R_2 bezpośrednio zależą od oszacowań podanych w lematach odpowiednio 2 i 3. Zaletą takiego testu jest fakt, iż problem uporządkowanego kolorowania drzew jest obliczeniowo łatwy, więc istnieje możliwość porównania wartości wspomnianych oszacowań z dokładną wartością uporządkowanego indeksu chromatycznego drzewa. Wyniki zostały zilustrowane w postaci wykresu. Dla każdej wartości rzędu grafu $n \in \{100,110,120,...,990,1000\}$ wygenerowano 100 drzew losowych. Następnie dla każdego drzewa obliczano wartość dokładną indeksu chromatycznego (za pomocą algorytmu

opisanego w [6]), wartość oszacowań z lematów 2 i 3, które oznaczono symbolami odpowiednio $B_1(\Delta,n)$ oraz $B_2(\Delta,n)$. Uśrednione wartości tych parametrów przedstawiono w postaci wykresu na rysunku 1.



Rys.1. Porównanie oszacowań oraz dokładnej wartości uporządkowanego indeksu chromatycznego w przypadku drzew losowych

4. ZAKOŃCZENIE

W pracy zdefiniowano funkcję dobroci algorytmu dla problemu MERST opisanego w [1]. Dokonano jej porównania z funkcją dobroci pochodzącą z [1]. Porównanie obejmowało analizę teoretyczną, którą podsumowuje twierdzenie 1 oraz doświadczenia komputerowe, których wyniki zebrano w poprzednim rozdziale. Podobnego porównania dokonano w przypadku oszacowań uporządkowanego indeksu chromatycznego drzew, pochodzących z prac [1] oraz [2], gdyż jakość tych oszacowań bezpośrednio wpływa na jakość odpowiednich funkcji dobroci. Warto jeszcze dodać, że jeśli $\Delta > 4$, to różnica pomiędzy wartościami oszacowań z lematów 2 oraz 3 może być dowolnie duża, gdyż

$$\begin{aligned}\frac{(\Delta-2)\lg n}{\lg\Delta-1}-\Delta\log_\Delta m &\geq \left(\frac{(\Delta-2)\lg\Delta}{\lg(\Delta/2)}-\Delta\right)\log_\Delta m \\ &= \left(\frac{\Delta-2}{\lg(\Delta/2)}\right)\log_\Delta m\end{aligned}$$

oraz $\Delta > 4$ implikuje

$$\frac{\Delta-2}{\lg(\Delta/2)}>0.$$

Dla ustalonej wartości stopnia grafu Δ powyższe wyrażenie jest stałe, co oznacza, że dla ustalonego parametru $\Delta > 4$, różnica pomiędzy porównywanymi oszacowaniami jest rzędu $\Omega(\lg m)$.

BIBLIOGRAFIA

[1] Makino K., Uno Y., Ibaraki T.: *On minimum edge ranking spanning trees*, J. Algorithms 38 (2001) 411-437.

[2] Dereniowski D.: *Uporządkowane kolorowanie krawędzi grafów dwudzielnych*. Raport Techniczny 31/2002 WETI PG, 2002.

[3] Iyer A.V., Ratliff H.D., Wijayan G.: *Parallel assembly of modular products - an analysis*, Tech. Report 88-06, Georgia Institute of Technology, 1988.

[4] Iyer A.V., Ratliff H.D., Wijayan G.: *On an edge ranking problem of trees and graphs,* Discrete Appl. Math. 30 (1991) 43-52.

[5] De la Torre P., Greenlaw R., Schäffer A.A.: *Optimal edge ranking of trees in polynomial time*, Algorithmica 13 (1995) 529-618.

[6] Lam T.W., Yue F.L.: *Optimal edge ranking of trees in linear time*, Proc. of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (1998) 436-445.

[7] Makino K., Uno Y., Ibaraki T.: *Minimum edge ranking spanning trees of threshold graphs*, LNCS 2518 (2002) 428-440.

[8] Fürer M., Raghavachari B.: *Approximating the minimum-degree Steiner tree to within one of optimal,* J. Algorithms 17 (1994) 409-423.

ANALYSIS OF AN APPROXIMATE ALGORITHM FOR THE MINIMUM EDGE RANKING SPANNING TREE PROBLEM

Summary

Edge k-ranking of a graph is a labeling of its edges with k colors such that each path between two edges having the same color contains an edge with a bigger color. The edge ranking number of a graph G is denoted by $\chi_r'(G)$ and defined as the smallest integer k such that there exists an edge k-ranking of G. In the minimum edge ranking spanning tree problem (MERST) we want to find a spanning tree T of a given graph G such that the value of $\chi_r'(T)$ is as small as possible. There exists a linear time algorithm for the edge ranking problem of trees [6]. However, problem MERST is NP-hard for general graphs [1]. Makino, Uno and Ibaraki gave a polynomial time approximate algorithm solving this problem. In this paper we use the bound proved in [2] to obtain a performance ratio of the algorithm, which is better than the ratio given in [1] for $\Delta^* > 3$.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Marcin Fuszara, Adrian Kosowski

Katedra Algorytmów i Modelowania Systemów, Politechnika Gdańska

STRATEGIE ALGORYTMICZNE RYWALIZACJI AUTONOMICZNYCH OBIEKTÓW MILITARNYCH NA PLATFORMIE DEDYKOWANEJ ROBOCODE

Streszczenie

Platformę Robocode opracowano w IBM jako środowisko symulacji walk wirtualnych robotów. Stosowane jest ono przez tysiące programistów na całym świecie do zabawy, nauki oraz badań nad mechanizmami sztucznej inteligencji i metodami uczenia. Artykuł rozpoczyna się krótką charakterystyką platformy i jej możliwości. Następnie autorzy na podstawie wiedzy zdobytej od innych programistów i doświadczeń wywiedzionych z licznych eksperymentów przeprowadzonych na robotach własnych i robotach innych osób przedstawiają charakterystykę starć robotów, identyfikując problemy napotykane podczas projektowania logiki robota i proponując sposoby ich rozwiązania. Najważniejszą część pracy stanowi przegląd niektórych podstawowych strategii używanych w logice robotów wraz ze wskazówkami dotyczącymi ich stosowania.

1. WSTĘP

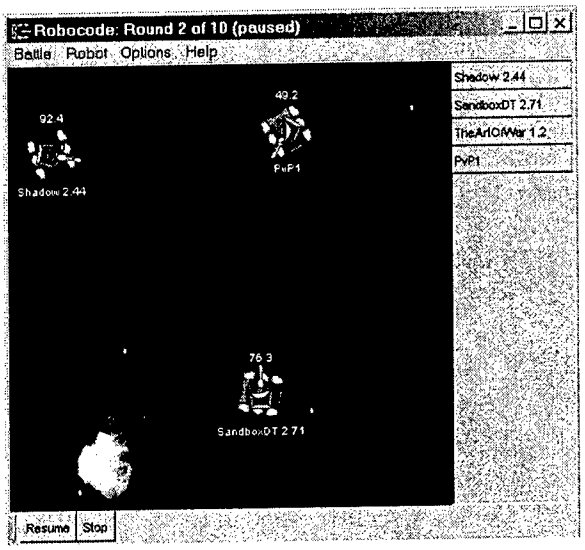
Robocode jest łatwym w użyciu, przyjaznym dla użytkownika środowiskiem symulacji walk wirtualnych robotów. Robotami są czołgi, dla których użytkownicy tworzą oprogramowanie sterujące logiką w języku Java. Roboty umieszczane są na prostokątnej dwuwymiarowej arenie i walczą aż do wyłonienia zwycięzcy lub zniszczenia wszystkich robotów.

Autorem Robocode jest Mathew Nelson z firmy IBM. Środowisko powstało jako gra dla programistów, lecz szybko dostrzeżono jego walory edukacyjne jako pomoc w nauczaniu języka Java. Szybko rosnącą popularność Robocode widać po ilości ogólnoświatowych lig i konkursów. Obecnie pasja programowania robotów skupia tysiące programistów testujących i dopracowujących zaawansowane algorytmy sztucznej inteligencji, które pozwolą ich robotom pokonać przeciwników i piąć się na szczyty rankingów.

2. CHARAKTERYSTYKA ŚRODOWISKA

Użytkownik korzystający z Robocode otrzymuje pełne środowisko zawierające narzędzia do gromadzenia, segregowania w pakiety i przeglądania stworzonych robotów, prosty edytor kodu źródłowego, kompilator oraz, co najważniejsze, system wizualizacji walk, za pomocą którego można w czasie rzeczywistym obserwować rozgrywane starcia.

Logika robotów implementowana jest poprzez stworzenie klasy dziedziczącej z klasy bazowej robota. Udostępnia ona zestaw metod, które pozwalają na uzyskiwanie informacji o stanie robota i sytuacji na polu walki oraz sterowanie elementami czołgu. Pełna dokumentacja API Robocode znajduje się na stronie domowej środowiska [1].



Rys.1. Środowisko Robocode

Ważną cechą Robocode, jedną z ważniejszych w tego typu środowiskach, jest sposób mierzenia czasu. Jest on odmierzany w klatkach (ang. *frames*). Obliczenia związane z logiką robotów wykonywane są pomiędzy klatkami. Wątek każdego z robotów biorących udział w walce ma ograniczony czas (ok. 20 milisekund) na wykonanie obliczeń, po czym musi podjąć decyzje i wykonać operacje sterujące czołgiem. W każdej klatce roboty wykonują żądane operacje, obliczana jest fizyka robotów, kolizje między nimi, kolizje z granicami pola walki oraz pociskami. Bardziej szczegółowy opis działania symulatora opisany jest w [2].

Czołgi mogą wykonywać dwa rodzaje operacji: blokujące i nieblokujące. Operacje blokujące powodują wykonanie żądanej czynności np. obrót, przemieszczenie o określoną liczbę jednostek, a wykonywanie wątku logiki robota jest wznawiane dopiero po zakończeniu wykonywania czynności. Dużo więcej możliwości dają operacje nieblokujące, które są traktowane jak rozkazy i mogą być przyjmowane co klatkę. Czołg wykonuje takie polecenia równolegle. Dzięki temu można mu rozkazać, aby jednocześnie poruszał się, zmieniał kierunek ruchu, skanował pole walki, obracał wieżyczką i wystrzeliwał pociski.

Informacje o otaczającym świecie robot uzyskuje poprzez zdarzenia. Za każdym razem, gdy w otoczeniu robota nastąpi określone zdarzenie, odpowiednia informacja jest generowana do kolejki zdarzeń robota, skąd może być następnie pobrana i przetworzona przez program logiki robota. Możliwe zdarzenia obejmują między innymi: uzyskanie na radarze obrazu innego robota, zderzenie z innym robotem, zderzenie ze ścianą, trafienie przez pocisk, trafienie przeciwnika pociskiem, trafienie pociskiem w ścianę, zniszczenie robota, zwycięstwo.

3. BUDOWA ROBOTA

Czołgi biorące udział w walkach Robocode składają się z podwozia oraz osadzonej na nim wieżyczki z działem i radarem. Wygląd takiego robota przedstawia rysunek 2.



Rys.2. Budowa robota

Podwozie robota może poruszać się do przodu i do tyłu ze skończoną maksymalną prędkością 8 jednostek/klatkę. Przyspieszenie i hamowanie także są skończone, więc robot rozpędza się i hamuje przez pewną liczbę klatek. Czołg skręca skończoną ilość stopni na klatkę i robi to tym wolniej im szybciej się porusza.

Działo osadzone na podwoziu i osadzony na dziale radar mogą obracać się niezależnie. Działo może wystrzeliwać pociski, które przemieszczają się z prędkością zależną od dobranej mocy pocisku. Mają one zasięg ograniczony jedynie granicami pola walki. Pociski poruszają się po liniach prostych i wybuchają w kolizji z robotem lub ścianą.

Istotny jest sposób działania radaru. Radar może wykonywać obroty. Położenia radaru w klatce bieżącej i klatce następnej tworzą kąt, który wraz z zasięgiem radaru wyznaczają pewien wycinek koła. Wszystkie pozostałe roboty znajdujące się na arenie w obrębie tego wycinka generują zdarzenie uzyskania obrazu robota, które może być przechwycone przez program i wykorzystane do uzyskania informacji o przeciwniku. Sposoby uzyskiwania informacji o otoczeniu omówione są w punkcie 5.1.

4. CHARAKTERYSTYKA STARCIA

4.1. Rodzaje walk

Walki robotów odbywają się na arenie o konfigurowalnych rozmiarach. Długość boku areny może być obrana w zakresie od 400 do 5000 jednostek. Na arenie w jednym momencie może znajdować się teoretycznie dowolna liczba robotów. Identyfikuje się trzy podstawowe rodzaje starć: pojedynki, walki wiele na wiele, walki drużynowe.

W pojedynku biorą udział dokładnie dwa roboty. W walce wiele na wiele na arenie umieszczana jest większa liczba robotów. Zwykle liczba ta waha się od kilku do kilkunastu. W każdym przypadku zadaniem robota jest jak najdłuższe przetrwanie na polu walki i zniszczenie przeciwników.

Walki drużynowe stanowią najbardziej złożony przypadek walk w Robocode. Drużynę stanowi pewna ilość czołgów, które posiadają mechanizmy komunikacji między sobą i mogą współpracować w celu pokonania drużyny przeciwnej. W drużynie mogą istnieć dwa rodzaje robotów: zwykły robot oraz droid. Droid różni się od zwykłego robota tym, że nie posiada radaru, w zamian za co otrzymuje więcej energii. Może to prowadzić do wykształcania się drużyn, w których jeden lub kilka robotów specjalizują się w uzyskiwaniu i przetwarzaniu informacji o polu walki, podczas gdy silne droidy zajmują się wykonywaniem rozkazów ataku lub obrony. Więcej informacji na temat walk drużynowych oraz przykładowe programy można znaleźć w [3].

4.2. Zarządzanie energią

Domyślnie na początku walki każdy robot umieszczany na arenie otrzymuje sto punktów energii. Energię tę robot traci w przypadku trafienia pociskiem, zderzenia ze ścianą lub zderzenia z innym robotem. Energia zużywana jest także podczas wystrzału pocisku. Robot może wystrzelić pocisk o mocy od 0.1 do 3.0 punktów energii. Silniejsze pociski poruszają się wolniej, ale też wyrządzają większe szkody. Trafienie przeciwnika, poza zmniejszeniem poziomu jego energii, powoduje trzykrotny zwrot energii zużytej na wystrzelenie pocisku. Ważny jest więc sposób dysponowania energią.

Gdy robot zużyje całą swoją energię, przechodzi on w stan wyłączony (ang. *disabled*), w którym nie może wykonywać żadnych operacji. Jeżeli któryś z jego wystrzelonych pocisków trafi przeciwnika, robot taki może odzyskać energię i znowu funkcjonować. Jeżeli robot bez energii zostanie trafiony, jest on niszczony i usuwany z pola walki.

Fakt zużywania energii na wystrzeliwanie pocisków jest ważny z punktu widzenia unikania trafienia przez przeciwnika, co omówione zostanie w punkcie 5.1.

5. PRZEGLĄD STRATEGII

W programie każdego robota można wyróżnić trzy podstawowe systemy, których działanie związane jest ze sterowaniem trzema elementami robota. Są to: skanowanie i gromadzenie informacji, sterowanie podwoziem i strategia poruszania, celowanie i strzelanie. Dla każdego z tych elementów logiki robota możliwe jest opracowanie i zamienne stosowanie różnorodnych strategii, które najlepiej sprawują się w danym rodzaju walki i w aktualnej sytuacji na arenie.

5.1. Gromadzenie informacji o otoczeniu

Podstawą dla wszelkich działań robota jest posiadanie aktualnych i wiarygodnych informacji o otoczeniu. Informacje te wiążą się głównie z położeniem i stanem przeciwników. W zdarzeniu generowanym w momencie znalezienia się przeciwnika w polu widzenia radaru robot otrzymuje informacje o położeniu przeciwnika, kierunku jego ruchu, prędkości i poziomie energii.

Ostatnia z tych informacji jest bardzo istotna dla potrzeb unikania pocisków przeciwnika. Ma to związek z tym, że radar nie przekazuje informacji o pociskach przemieszczających się po polu walki. Dlatego jedynym sposobem na wykrycie faktu wystrzelenia pocisku przez przeciwnika jest monitorowanie spadku jego energii. Wystrzelenie pocisku wiąże się ze zużyciem energii robota, więc zaobserwowanie u przeciwnika drobnego spadku energii w dwóch sąsiednich klatkach może oznaczać, że przeciwnik wystrzelił pocisk. Dlatego tak ważna jest aktualność danych o przeciwnikach.

W przypadku walk jeden na jednego możliwe jest stałe śledzenie swojego przeciwnika i uzyskiwanie aktualnych informacji w każdej klatce. Radar może w jednej klatce obrócić się aż o 45 stopni, więc utrzymanie przeciwnika na radarze nie jest problemem nawet, gdy się on szybko porusza. W walkach, w których bierze udział więcej robotów, należy zadbać, aby obszar obserwowany przez radar był jak najmniejszy i jednocześnie obejmował maksymalną liczbę robotów. Sposób minimalizacji przebiegu radaru wraz z przykładowym kodem źródłowym opisany jest w [4].

5.2. Celowanie

Na podstawie danych zgromadzonych na temat przeciwnika możliwe jest określenie aktualnej i przewidzenie przyszłej pozycji robota. Głównym problemem w celowaniu jest fakt, iż przeciwnik może wykrywać pociski i próbować ich unikać albo nawet może poruszać się w sposób losowy niemożliwy do przewidzenia.

5.2.1. Celowanie czołowe

Jest to najprostszy sposób celowania polegający na wystrzeleniu pocisku w miejsce, w którym ostatnio zaobserwowano przeciwnika. Ten sposób wydaje się być naiwny, lecz działa zaskakująco dobrze w przypadku niektórych robotów, próbujących unikać pocisków poprzez ruchy oscylacyjne.

5.2.2. Predykcja liniowa i symulacja ruchu przeciwnika

Najprostszy sposób przewidywania pozycji robota polega na ekstrapolacji jego ruchu zarejestrowanego w dwóch ostatnich klatkach. Metoda dobrze sprawuje się dla robotów poruszających się po prostych. Zawodzi jednak dla przeciwników poruszających się po łukach.

Dla takich przeciwników dobre wyniki daje bardziej uniwersalna metoda symulacji ruchu przeciwnika. Sposób ten bierze pod uwagę nie tylko położenie przeciwnika i jego prędkość, ale także zmianę kierunku ruchu w kilku ostatnich klatkach. Na tej podstawie szacuje się, czy robot zwalnia, czy też przyspiesza i jak silnie skręca. Stosując przyjęte założenia symuluje się ruch przeciwnika w taki sposób, w jaki robi to sam silnik symulacyjny Robocode i robi się to przez taką liczbę klatek, która pozwoli pociskowi na dotarcie do przeciwnika w przewidzianej chwili czasu.

5.2.3. Wyszukiwanie wzorców

Wyszukiwanie wzorców (ang. *pattern matching*) jest jedną z prostszych w implementacji metod uczenia, zmierzającą do wyszukiwania powtarzalnych schematów zachowań w ruchach przeciwnika. Zakłada ona, że przeciwnik powtarza wykonywane ruchy, co jest zwykle prawdą nawet w przypadku dość zaawansowanych robotów unikających pocisków.

Metoda ta polega na rejestrowaniu w każdej klatce reprezentatywnej porcji informacji o ruchu przeciwnika. Zwykle rejestrowane są: prędkość (v) oraz zmiana kierunku ruchu w stosunku do poprzedniej klatki (a). W ten sposób powstaje ciąg par (v_0, a_0), (v_1, a_1),..., (v_n, a_n) gdzie n to liczba zarejestrowanych rekordów. Predykcja jest zwykle trafna już dla kilkuset rekordów.

Wyszukiwanie wzorca polega na wyszukaniu w zarejestrowanej historii części ciągu, która najlepiej pasuje do ostatnich k zarejestrowanych par, gdzie k jest pewną obraną stałą (zwykle rzędu kilkunastu lub kilkudziesięciu). W tym celu wylicza się oceny:

$$E_m = \sum_{i=0}^{i=k-1} \left((v_{m+i} - v_{n-k+i})^2 + (a_{m+i} - a_{n-k+i})^2 \right) \tag{5.1}$$

dla $0 \le m \le n-2k$

Następnie wybiera się taką wartość m, dla której wartość błędu E_m jest najmniejsza i jeżeli nie przekroczy ona pewnego progu, traktuje się wartość m jako indeks początku wyszukanego wzorca. Zakładając, że przeciwnik powtórzy wykonany wcześniej ruch, na podstawie tego wzorca i przebiegu historii ruchu po nim zarejestrowanego można przewidzieć prawdopodobną pozycję przeciwnika w przyszłości i w to miejsce posłać pocisk.

5.3. Poruszanie

W logice poruszania się robota należy uwzględnić konieczność unikania zderzeń ze ścianami i innymi robotami. W pojedynkach wystarczy utrzymywać odpowiednią odległość od przeciwnika zależną od tego, czy robot przyjmuje strategię ofensywną czy defensywną. W walce dla wielu robotów w grę wchodzi konieczność wyszukiwania najbezpieczniejszych miejsc na arenie i znajdowania sposobów na dotarcie do nich.

Zaawansowane roboty próbują przewidzieć momenty, w których przeciwnicy wystrzeliwują pociski oraz kierunki strzałów, po czym poruszają się w sposób minimalizujący ryzyko trafienia.

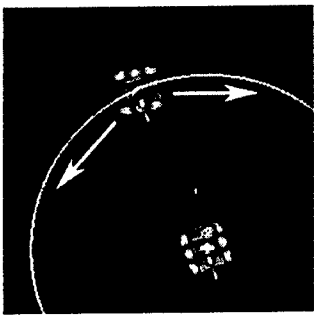
W dalszej części podrozdziału opisano niektóre strategie poruszania się dla pojedynków i walk z udziałem większej liczby robotów.

5.3.1. Unikanie pocisków

W punkcie 5.1 podano sposób wykrycia wystrzelenia pocisku przez przeciwnika. Fakt, iż nieznany jest kąt obrotu wieżyczki przeciwnika, uniemożliwia uzyskanie wiedzy pewnej na temat pocisków poruszających się po polu walki. Jedyną metodą jest zgadywanie. W momencie wystrzelenia pocisku przez przeciwnika robot, stosując różne metody celowania, może przewidzieć zestaw kątów, pod jakimi przeciwnik mógł wystrzelić pocisk. Następnie w zależności od przyjętej strategii poruszania, robot powinien wykonać ruchy zmierzające do uniknięcia pocisków we wszystkich przewidzianych trajektoriach lotu, lub tylko niektórych z nich w zależności od prawdopodobieństwa trafienia. Prawdopodobieństwa te mogą być wyznaczone w drodze analizy statystycznej pocisków wystrzelonych przez przeciwnika wcześniej w danej walce. Dobry robot uczący się potrafi skutecznie unikać trafień niezależnie od metody celowania używanej przez przeciwnika.

5.3.2. Poruszanie centryczne

W walkach jeden na jednego dobrą strategią poruszania jest poruszanie centryczne. Polega ono na poruszaniu się po łukach wokół pewnego punktu, najczęściej przeciwnika lub środka pola walki. Poruszanie się w ten sposób wymusza prostopadłe ustawienie podwozia do przeciwnika, co przyspiesza usuwanie się z toru lotu pocisku przeciwnika, a częste zmiany kierunku w ruchu po łuku utrudniają trafienie. Poza tym ruch ten ułatwia utrzymywanie określonego dystansu do przeciwnika.



Rys. 3. Poruszanie centryczne

Inną odmianą poruszania centrycznego jest poruszanie się wokół środka areny. Stosują ją roboty defensywne, utrzymujące dużą odległość do przeciwnika. Poruszają się po łukach

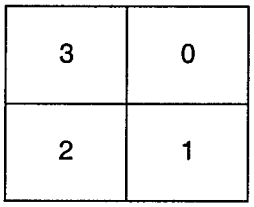
po przeciwnej stronie środka pola walki jedynie w zakresie kątów zapewniających dużą odległość do przeciwnika.

5.3.3. Poruszanie losowe

Większość udanych robotów stosuje dość złożone metody uczenia się ruchów przeciwnika. W starciu z takim przeciwnikiem nawet najlepsza strategia poruszania może być rozpoznawana przez przeciwnika. Roboty unikające pocisków zwykle robią to w charakterystyczny sposób, łatwy do znalezienia nawet w prostym systemie wyszukiwania wzorców.

Dlatego dość skutecznym sposobem unikania trafienia jest poruszanie się w losowy sposób. Duża losowość ruchu utrudnia przeciwnikowi analizę i zmniejsza prawdopodobieństwo precyzyjnego trafienia. Losowość można wprowadzić do różnego rodzaju ruchów, w tym poruszania centrycznego.

Dla potrzeb robota *PvP1* autor opracował strategię poruszania nazwaną *sector movement*. Jej podstawą jest podział pola walki na cztery sektory zgodnie z rysunkiem 4.



Rys. 4. Podział pola walki w *sector movement*

Przyjmując, że przeciwnik znajduje się w sektorze *s*, robot zawsze zmierza w kierunku sektora znajdującego się po drugiej stronie areny, czyli *(s+2) mod 4*. Pomiędzy sektorami robot porusza się po losowej łamanej. Gdy robot znajduje się wewnątrz wymaganego sektora, porusza się, losując po kolei punkty wewnątrz sektora, do których ma dotrzeć. Ze względu na niewielką objętość sektora na typowych arenach losowy ruch wewnątrz sektora daje krótkie okresy poruszania liniowego i utrudnia trafienie przez przeciwnika.

5.3.4. Poruszanie antygrawitacyjne

Poruszanie antygrawitacyjne (ang. *antigravity movement*) jest obecnie najskuteczniejszą metodą poruszania w walkach z dużą ilością robotów na polu walki. Polega ono na rozmieszczeniu na polu walki wirtualnych pól siłowych, oddziałujących na robota. Pola te mogą przyciągać robota lub go odpychać w zależności od przyporządkowanej wagi. Każde źródło pola posiada siłę *(p)* dodatnią lub ujemną oraz współczynnik odległości *(f)*. Mając dany zestaw źródeł, można wyliczyć wartość *F* wypadkowej siły działającej na robota:

$$F = \sum_i \frac{p_i}{d_i^f} \tag{5.2}$$

gdzie d_i jest odległością źródła pola od robota.

Domyślnie $f=2$, jednak może być większe dla punktów grawitacyjnych, które mają działać tylko na małych odległościach. Siłę oblicza się jako wektor wyliczając oddzielnie składowe dla każdego wymiaru. Robot porusza się zgodnie z kierunkiem i zwrotem siły działającej na niego.

Jako pola grawitacyjne stosuje się inne roboty jako punkty tak, aby utrzymywać do nich odpowiedni dystans oraz ściany jako proste, dając im wysokie współczynniki f, aby odpychały robota tylko w pobliżu brzegów pola walki. Poza tym do zbioru punktów grawitacyjnych dodaje się przewidywane pozycje wrogich pocisków, dzięki czemu robot unika trafienia. Możliwe jest ustanowienie dodatniego pola w centrum areny, aby robot starał się utrzymywać środek pola walki w swoim panowaniu, albo losowo dodawać punkty w różnych miejscach areny w celu urozmaicenia ruchu robota i dodania odrobiny losowości. Metoda poruszania antygrawitacyjnego daje wiele możliwości. Przykład prostej implementacji tej metody znajduje się w [5].

5.4. Metody statystyczne

Jednym ze sposobów wspomagania uczenia się robota jest prowadzenie statystyk i ich analiza. Statystyki mogą obejmować ilości trafień i chybień dla każdej użytej przez robota metody celowania, skuteczność metod unikania pocisków, statystyki zwycięstw i przegranych przy użyciu danej strategii itp.

Analiza statystyk może na przykład pomóc w korekcji kąta strzału w celu zwiększenia prawdopodobieństwa trafienia. Wybór metod celowania i ruchu oraz sposobów unikania pocisków także można oprzeć na statystykach.

6. ZASTOSOWANIA DYDAKTYCZNE PLATFORMY ROBOCODE

Pisanie programów dla środowiska Robocode jako nauka połączona z zabawą czyni tę platformę doskonałym narzędziem do zastosowań dydaktycznych, zarówno w nauczaniu języka Java jak i podstawowych mechanizmów sztucznej inteligencji. Fakt ten dostrzeżony już został przez wielu nauczycieli akademickich na całym świecie.

Koło Naukowe Sfera, którego członkami są autorzy tej pracy, zorganizowało pod opieką pana dr inż. Michała Małafiejskiego na wydziale Elektroniki Telekomunikacji i Informatyki Politechniki Gdańskiej turniej robotów pod nazwą I Otwarty Turniej Robocode. Obok studentów wydziału wzięli w nim udział także uczniowie gimnazjów i liceów z Trójmiasta. Turniej był dla uczestników silną motywacją do opanowania języka Java oraz zgłębienia tajników uczenia się sztucznej inteligencji.

W finale turnieju wzięło udział sześć robotów o nazwach: Kasei 1.2, HuntaKilla 1.0, Masochista 1.0, Pandemonium 1.0, PvP1 IOTR oraz Robbo 1.0.0. Poniżej przedstawiona jest krótka charakterystyka każdego z nich.

- Kasei jest robotem, który bada ogólną strategię ruchu przeciwnika i na tej podstawie modyfikuje swoją. Porusza się w mało przewidywalny sposób po łukach, więc trudno go trafić. Celuje jednak w sposób czołowy, co czyni go niezbyt skutecznym.
- HuntaKilla jest prostym robotem poruszającym się po okręgach, dzięki czemu jest odporny na celowanie z predykcją liniową. Celuje czołowo. Mimo swojej prostoty okazał się w turnieju wyjątkowo udany.
- Masochista ma dobre działo z przewidywaniem liniowym. Bierze przy tym pod uwagę kilka ostatnich klatek, przewidując przyszłą prędkość przeciwnika. Porusza się po wylosowanym na początku walki grafie, co czyni jego ruch mało przewidywalnym, o ile w swoim ruchu nie wpadnie w cykl.
- Pandemonium porusza się centrycznie wokół przeciwnika. Celuje czołowo.

- PvP1 jest robotem opracowanym przez jednego z autorów publikacji. Zaimplementowano w nim część opisanych tu mechanizmów. Robot ten stosuje dwie strategie poruszania: *sector movement* oraz poruszanie centryczne wokół środka planszy przy utrzymaniu dużej odległości od przeciwnika. Strategie te przełączane są losowo co kilka sekund w celu utrudnienia wyuczenia się ruchu przez przeciwnika. PvP1 posiada precyzyjne działo z algorytmem celowania opartym o wyszukiwanie wzorców. Silne mechanizmy i dopracowanie uczyniło go najsilniejszym robotem turnieju.
- Robbo jest bardzo mocnym robotem, posiadającym zaawansowany mechanizm uczenia ruchów przeciwnika i celowania. Porusza się centrycznie wokół przeciwnika utrzymując małą, ofensywną odległość do przeciwnika. Proste reguły poruszania czyniły go zbyt łatwym celem dla PvP1

Finał turnieju odbywał się w systemie pucharowym. Programy zostały podzielone na dwie grupy po trzy roboty, w których miały miejsce walki każdego robota z każdym. Każdy mecz trwał pięć rund. Oto wyniki walk w grupach:

Grupa A		Grupa B	
HuntaKilla - Masochista	5:0	Pandemonium - Kasei	2:3
Masochista - PvP1	0:5	Pandemonium - Robbo	2:3
PvP1 - HuntaKilla	5:0	Robbo - Kasei	5:0

W półfinale walczyły: pierwszy robot z grupy A z drugim robotem z grupy B oraz drugi robot z grupy A z pierwszym robotem z grupy B. Mecze trwały po dziesięć rund. Oto wyniki tych starć:

Kasei - PvP1	0:10
Robbo - HuntaKilla	8:2

Zwycięskie roboty z półfinałów wzięły udział w finale piętnastorundowym. Poza tym odbyły się dwie dziesięciorundowe walki o trzecie i piąte miejsce. Poniżej znajdują się wyniki:

Walka o 1. miejsce	Robbo - PvP1	4:11
Walka o 3. miejsce	HuntaKilla - Kasei	7:3
Walka o 5. miejsce	Masochista - Pandemonium	2:3

Ostatecznie, kolejne miejsca w turnieju zajęły roboty: PvP1, Robbo, HuntaKilla, Kasei, Pandemonium, Masochista. Turniej pokazał silną przewagę robotów wykorzystujących mechanizmy uczenia. Bardzo ważne okazało się dopasowanie strategii ruchu i celowania oraz staranne dopracowanie algorytmów. Dzięki tym cechom wyraźną dominację w czasie całego turnieju przejawiał PvP1.

Koło Naukowe Sfera planuje organizację większej ilości tego typu turniejów. Niemałe zainteresowanie konkursem rokuje szanse na rozpowszechnienie się platformy Robocode jako narzędzia stosowanego w edukacji.

7. PODSUMOWANIE

Przedstawione propozycje rozwiązań logiki robota stanowią jedynie niewielki wybór opracowanych już metod i to w najprostszej postaci. Jednak wyniki przeprowadzonych zawodów pokazują, że implementacja podanych strategii może skutkować powstaniem

całkiem silnego robota. W pozycjach [6] i [7] czytelnik znajdzie mnóstwo informacji, opracowań i wskazówek na temat sposobów walki robotów w Robocode i doboru odpowiednich elementów logiki robota. Niektóre obszary, takie jak walka drużynowa, nadal używają niedoskonałych metod i pozostawiają mnóstwo miejsca dla rozważań i eksperymentów.

BIBLIOGRAFIA

[1] IBM alphaWorks: *Robocode Home Page*
http://robocode.alphaworks.ibm.com/
[2] Sing Li: *Rock'em, sock'em Robocode!*
http://www-106.ibm.com/developerworks/java/library/j-robocode/
[3] Sing Li: *Rock'em, sock'em Robocode: Round 2*
http://www-106.ibm.com/developerworks/library/j-robocode2/
[4] Eivind Bjarte Tjore: *Secrets from the Robocode masters: Radar sweeps*
http://www-106.ibm.com/developerworks/library/j-radar.html
[5] Alisdair Owens: *Secrets from the Robocode masters: Anti-gravity movement*
http://www-106.ibm.com/developerworks/library/j-antigrav/
[6] *Robo Wiki*
http://robowiki.dyndns.org/
[7] *Secrets from the Robocode masters*
http://www-106.ibm.com/developerworks/library/j-robotips/

ALGORYTHMIC STRATEGIES FOR BATTLES OF AUTONOMOUS MILITARY OBJECTS IN THE ROBOCODE DEDICATED SYSTEM

Summary

The Robocode system was created by IBM as a robotics battle simulation environment. It is used by thousands of programmers all around the world to play, learn and to conduct research on artificial intelligence and machine learning methods. This article begins with a description of the environment and its capabilities. Then the authors based on knowledge gained from other programmers and experience coming from numerous experiments carried out on their own robots and robots made by other people present the profile of robot battles and identify problems encountered during design of robot logic as well as propose their solutions. The most important part of this paper is a survey of some basic strategies used in Robocode robot logic giving hints on their use.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Robert Janczewski

Katedra Algorytmów i Modelowania Systemów, Politechnika Gdańska

UOGÓLNIONE ALGORYTMY ZACHŁANNE W KONTRASTOWYM KOLOROWANIU GRAFÓW

Streszczenie

Kontrastowe kolorowanie grafów jest uogólnieniem wierzchołkowego kolorowania grafów znajdującym zastosowanie m.in. w problemie przydziału częstotliwości, zagadnieniu komiwojażera, układaniu rozkładów zajęć i szeregowaniu zadań. Niniejszy referat poświęcony jest uogólnionym algorytmom zachłannym, tj. algorytmom, które kolorują kontrastowo grafy w oparciu o reguły stanowiące uogólnienie reguły zachłannego kolorowania. Referat zawiera opis tych algorytmów, krótką analizę ich właściwości oraz wyniki eksperymentów komputerowych przeprowadzonych w celu sprawdzenia, o ile są te algorytmy lepsze od zachłannych.

1. WSTĘP

Kontrastowe kolorowanie grafów powstało jako matematyczny model dla tzw. problemu przydziału częstotliwości, po czym przekształciło się w niezależny dział teorii grafów, który znajduje zastosowanie w wielu innych zagadnieniach takich jak problem komiwojażera, układanie rozkładów zajęć i szeregowanie zadań. Problem przydziału częstotliwości [1] definiuje się zazwyczaj następująco: na pewnym obszarze znajduje się grupa nadajników radiowych, z których część może z sobą interferować; trzeba przydzielić nadajnikom częstotliwości w taki sposób, by nie zakłócały się w trakcie nadawania i aby szerokość wykorzystanego pasma częstotliwości, tzn. różnica pomiędzy największą a najmniejszą z wykorzystanych częstotliwości, była minimalna.

Kontrastowe kolorowanie grafów jest uogólnieniem klasycznego wierzchołkowego kolorowania grafów. Przypomnijmy, że jeżeli G jest grafem, którego zbiorem wierzchołków jest V, a zbiorem krawędzi E, to wierzchołkowym pokolorowaniem grafu G nazywa się każdą funkcję, która przyporządkowuje wierzchołkom grafu G liczby całkowite (kolory) w taki sposób, że wierzchołki sąsiadujące otrzymują różniące się kolory. T-pokolorowaniem (kontrastowym pokolorowaniem) grafu G nazywamy każdą funkcję, która przyporządkowuje wierzchołkom grafu G kolory w taki sposób, że odległość kolorów (tzn. moduł z ich różnicy) przydzielonych sąsiadującym wierzchołkom nie należy do ustalonego zbioru T (zbiór T może być tutaj dowolnym skończonym podzbiorem zbioru liczb całkowitych nieujemnych, zawierającym w sobie 0; widać, że jeżeli jedynym elementem zbioru T jest 0, to T-pokolorowania i wierzchołkowe pokolorowania stają się synonimami).

Problem kontrastowego kolorowania grafów formułuje się zazwyczaj następująco: dany jest graf G i zbiór T; wyznacz takie T-pokolorowanie grafu G, którego rozpiętość, tzn. różnica pomiędzy największym, a najmniejszym wykorzystanym kolorem, jest minimalna. Kontrastowe kolorowanie grafów jest zagadnieniem o dużej złożoności obliczeniowej. I tak, problem ten jest silnie NP-trudny dla grafów pełnych, podkubicznych i regularnych (dla porównania: wierzchołkowe kolorowanie grafów jest NP-trudne tylko w ostatnim przypadku; pierwsze dwa można rozwiązać przy pomocy algorytmów o liniowej złożoności obliczeniowej) oraz wielu innych klas grafów [3, 6]. Z tego też powodu w praktyce stosuje się algorytmy, które mają niską złożoność obliczeniową i znajdują przybliżone rozwiązania tego problemu. Jednym z takich algorytmów jest algorytm zachłanny, będący tematem niniejszego referatu.

2. ALGORYTM ZACHŁANNY I JEGO UOGÓLNIENIA

Algorytm zachłanny stosowany w kontrastowym kolorowaniu grafów jest uogólnieniem algorytmu kolorowania zachłannego stosowanego w klasycznym kolorowaniu grafów. Niech G będzie grafem, który chcemy pokolorować kontrastowo, a T zbiorem, o którym była mowa w definicji T-pokolorowania. Algorytm zachłanny zaczyna od ponumerowania wierzchołków grafu G (istnieją różne sposoby numerowania wierzchołków prowadzące do różnych wariantów algorytmu zachłannego, takich jak T-SL, T-LF czy też T-DSATUR [2, 3, 5]). Następnie pierwszy wierzchołek koloruje kolorem 0, a każdy następny najmniejszym dopuszczalnym kolorem, tj. takim kolorem, którego odległość od kolorów przydzielonych wcześniej sąsiadom tego wierzchołka nie należy do zbioru T. Dla przykładu, jeżeli $T = \{0, 2, 3\}$, to kolorując 3-wierzchołkowy graf pełny K_3 algorytm zachłanny skorzysta z kolorów 0, 1 i 5.

Własności algorytmu zachłannego są stosunkowo dobrze znane. Wiadomo, że jego złożoność obliczeniowa jest rzędu $O(n^2|T|)$ oraz że nie zawsze daje w wyniku optymalne, tj. posiadające najmniejszą rozpiętość T-pokolorowanie (najmniejszym grafem, na którym algorytm zachłanny może się pomylić, jest 3-wierzchołkowy graf pełny). Wiadomo także, że wierzchołkowi v algorytm zachłanny przypisze kolor nie większy niż stopień tego wierzchołka (stopień wierzchołka = liczba wierzchołków, z którymi sąsiaduje) razy liczba elementów zbioru T, skąd wynika m.in. to, że rozpiętość pokolorowania wygenerowanego przez algorytm zachłanny nie przekracza iloczynu stopnia grafu (stopień grafu = maksymalny ze stopni jego wierzchołków) i liczby elementów zbioru T.

Zasadę, na której opiera się działanie algorytmu zachłannego można uogólniać na wiele sposobów. Nas będzie interesować następujące uogólnienie. Niech k będzie liczbą naturalną. Kolorując graf G, zaczynamy od ponumerowania jego wierzchołków. Pierwsze k z nich kolorujemy tak, by powstało optymalne pokolorowanie. Następnie, każdą kolejną grupę k wierzchołków kolorujemy tak, by użyte były tylko dopuszczalne kolory i aby powstało pokolorowanie częściowe o minimalnej rozpiętości. Algorytm działający zgodnie z tą zasadą nazywać będziemy k-zachłannym; widać, że algorytm zachłanny = algorytm 1-zachłanny.

Algorytmy k-zachłanne (dla $k > 1$) będą działały wolniej niż algorytm zachłanny, gdyż można bez trudu sprawdzić, że algorytm k-zachłanny działa w czasie $O(n^{k+1}|T|^k)$. Co więcej, ten algorytm także nie musi generować optymalnych T-pokolorowań, chociaż łatwo jest zauważyć, że algorytm k-zachłanny nie myli się na grafach posiadających mniej niż $k+1$ wierzchołków. Aby sprawdzić, czy i o ile algorytmy k-zachłanne są lepsze od zachłannego, poddano je opisanym dalej testom komputerowym.

3. TESTY KOMPUTEROWE

Aby zbadać, czy algorytmy k-zachłanne są lepsze od zachłannego, poddano je całej serii testów komputerowych. W ramach takiego testu wykonywano algorytm zachłanny i 2-zachłanny na tysiącu grafów losowych (o ustalonej liczbie wierzchołków n i ustalonym prawdopodobieństwie wylosowania krawędzi p) oraz wszystkich zbiorach T o maksymalnym elemencie mniejszym lub równym 6; następnie rozpiętości uzyskanych pokolorowań uśredniono, zaokrąglono do pierwszego miejsca po przecinku i umieszczono w poniższych tabelach.

Tabela 3.1

Wyniki testów dla algorytmu zachłannego

n/p	10	20	30	40	50	60	70	80	90
10	3	5	6,7	8,4	10	11,8	13,7	16,3	20
20	6,1	9,2	12,3	15,1	18,3	21,4	25,3	30,2	37,3
30	8,2	12,8	16,9	20,9	25,2	30,1	35,5	42,6	52,7
40	10,2	15,6	20,9	26,2	31,5	38,1	45	54,1	67,4
50	12,2	18,8	24,7	31,1	37,8	45,2	53,7	64,8	81,4

Tabela 3.2

Wyniki testów dla algorytmu 2-zachłannego

n/p	10	20	30	40	50	60	70	80	90
10	3,1	5,1	6,9	8,7	10,4	12,3	14,3	17	20,8
20	6,4	9,8	12,9	16,3	19,4	22,7	26,8	31,6	38,9
30	9	13,7	18,2	22,6	27,2	32,2	37,9	44,9	55
40	11,3	17,3	22,9	28,5	34,3	40,8	48,3	57,4	70,7
50	13,1	20,3	27,1	33,8	41,1	48,8	57,6	69,2	85,3

Jak nietrudno zauważyć, algorytm zachłanny okazał się lepszy we wszystkich badanych przypadkach, przy czym różnica pomiędzy badanymi algorytmami nie jest duża (rzędu kilku procent) i rośnie wraz ze wzrostem gęstości grafu. Innymi słowy, algorytm 2-zachłanny zachowuje się najlepiej dla grafów rzadkich; im graf jest gęstszy, tym gorsze są rezultaty uzyskiwane przez algorytm 2-zachłanny. Dokładniejsza analiza (przejrzenie i porównanie wyników uzyskanych na poszczególnych grafach przez oba algorytmy, a nie tylko porównanie średniej) pokazuje jednak, że nie zawsze algorytm zachłanny generuje lepsze wyniki niż 2-zachłanny. Warto zatem zbadać, jak zachowuje się poprawiony algorytm 2-zachłanny, tzn. algorytm, który najpierw koloruje graf zachłannie, a potem 2-zachłannie i jako wynik swojego działania zwraca lepsze z uzyskanych pokolorowań. Wyniki testów, którym poddano ten algorytm znajdują się w poniższej tabelce. Pokazują one, że poprawiony algorytm 2-zachłanny jest o kilkanaście procent lepszy od algorytmu zachłannego dla grafów rzadkich i o kilka procent lepszy dla grafów gęstych.

Tabela 3.3

Wyniki testów dla poprawionego algorytmu 2-zachłannego

n/p	10	20	30	40	50	60	70	80	90
10	2,7	4,7	6,4	8	9,6	11,5	13,4	16	19,9
20	5,8	8,8	11,8	14,6	17,8	20,9	24,8	29,6	36,8
30	7,9	12,5	16,4	20,5	24,6	29,5	34,9	41,9	52,1
40	9,9	15,2	20,5	25,8	31	37,6	44,4	53,3	66,6
50	10,6	18,5	24,3	30,6	37,4	44,6	53	64,1	80,5

6. PODSUMOWANIE

Zdefiniowane w treści artykułu algorytmy k-zachłanne działają lepiej od zachłannego dla małych oraz rzadkich grafów (w niektórych przynajmniej przypadkach). Im większa jest gęstość grafu, tym mniejszy jest potencjalny zysk płynący z zastąpienia algorytmu zachłannego k-zachłannym. Poprawione algorytmy k-zachłanne działają w czasie porównywalnym z k-zachłannymi (ich złożoności obliczeniowe mają to samo tempo wzrostu), a są lepsze o kilka do kilkunastu procent od zachłannego, przy czym im większa gęstość kolorowanego grafu, tym mniejszy zysk płynący z zastąpienia algorytmu zachłannego poprawionym k-zachłannym.

BIBLIOGRAFIA

[1] Hale W.K.: *Frequency assignment: theory and applications*, Proceedings of the IEEE 68 (1980), str. 1497-1514.

[2] Janczewski R., Kubale M.: *The T-DSATUR algorithm - an interesting generalization of the DSATUR algorithm*, Advanced Computer Systems V, Szczecin 1998, str. 288-289.

[3] Janczewski R.: *Kontrastowe kolorowanie grafów i jego zastosowania*, Rozprawa doktorska, 2001.

[4] Janczewski R.: *Divisibility and T-span of graphs*, Discrete Mathematics 234 (2001), str. 171-179.

[5] Janczewski R., Małafiejski M.: *T-SL, T-SLF i T-DSATUR - nowe heurystyki dla problemu przydziału częstotliwości*, Zeszyty Naukowe Politechniki Śląskiej, Seria Automatyka 136 (2002), str. 97-105.

[6] Giaro K., Janczewski R., Małafiejski M.: *The complexity of the T-coloring problem for graphs with small degree*, Discrete Applied Mathematics 129 (2003), str. 361-369.

GENERALIZED GREEDY ALGORITHMS AND T-COLORING OF GRAPHS

Summary

The paper is devoted to the so-called generalized greedy algorithms and their application to T-coloring. T-coloring of graphs is a generalization of classical vertex coloring of graphs whose appplications include: frequency assignment problem, travelling salesman problem, timetabling and scheduling. The generalized greedy algorithms are briefly discussed and compared to the greedy algorithm.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Krzysztof Kwarciany, Zbigniew Kokosiński

Wydział Inżynierii Elektrycznej i Komputerowej, Politechnika Krakowska

RÓWNOLEGŁE ALGORYTMY GENETYCZNE W ZASTOSOWANIU DO PROBLEMU KOLOROWANIA WIERZCHOŁKÓW GRAFU

Streszczenie

W niniejszej pracy drogą symulacji komputerowej przebadano równoległe algorytmy genetyczne (PGA) w zastosowaniu do problemu kolorowania wierzchołków grafu (GCP). Spośród kilku modeli PGA różniących się sposobem zrównoleglenia obliczeń w algorytmie genetycznym, liczbą procesorów i strukturą ich połączeń, liczbą i rodzajem koewolujących populacji, sposobem wymiany informacji genetycznej pomiędzy populacjami itp. wybrano dwa: zrównoleglenie globalne (model master–slave) oraz statyczne podpopulacje z migracją (model wyspowy). Przeprowadzono analizę symulowanej pracy wybranych równoległych algorytmów genetycznych z nowymi operatorami krzyżowania, wykazując eksperymentalnie ich efektywność. Szczegółowe wnioski sformułowane w tej pracy mogą być pomocne przy projektowaniu nowych równoległych algorytmów genetycznych dla problemu GCP.

1. WSTĘP

Algorytmy genetyczne są iteracyjną metodą optymalizacji opartą na mechanizmach zapożyczonych z ewolucji naturalnej. Łączą one ewolucyjną zasadę przeżycia najlepiej przystosowanych osobników w danej populacji z systematyczną, choć zrandomizowaną wymianą informacji pomiędzy kolejnymi populacjami. W każdym pokoleniu dzięki operacjom krzyżowania, mutacji i selekcji powstaje nowy zespół sztucznych organizmów (ciągów kodowych) reprezentujących zbiór rozwiązań problemu optymalizacyjnego. Pomimo wbudowanych mechanizmów losowości, algorytmy genetyczne nie sprowadzają się do zwykłego błądzenia przypadkowego, ale wykorzystują proces ewolucji do efektywnego przeszukiwania przestrzeni rozwiązań.

Modele równoległe GA umożliwiają zarówno zwiększenie szybkości działania, jak i pozwalają przeszukiwać jednocześnie dużo większe przestrzenie rozwiązań niż to miało miejsce przy przetwarzaniu sekwencyjnym [1,2].

Badania wybranych aspektów równoległych algorytmów genetycznych opisane w tej pracy przeprowadzono z wykorzystaniem oryginalnego programu symulacyjnego *PGA_for_GCP*, przeznaczonego do rozwiązywania problemu kolorowania wierzchołków

grafu (GCP) [15]. Program umożliwia wybór modelu równoległego algorytmu genetycznego oraz wielu istotnych dla jego dostrojenia parametrów.

W modelu zrównoleglenia globalnego zbadano dla przykładowych przypadków problemu GCP zależność czasu obliczeń algorytmu od ilości procesorów „slave", wyznaczając optymalną, ze względu na czas obliczeń w systemie, liczbę procesorów.

W modelu wyspowym przebadano wpływ na wyniki obliczeń schematu migracji oraz odstępu pomiędzy migracjami. Zbadano także model równoległy, w którym wyspy są izolowane.

Obok znanych operatorów mutacji porządkowej i mutacji First Fit [12], przebadano po raz pierwszy dwa oryginalne operatory krzyżowania SPPX (*Sum-Product Partition Crossover*) i CEX (*Conflict Elimination Crossover*) [11,12,15], wykazując ich dużą skuteczność w rozwiązywaniu problemu GCP.

2. PROBLEM KOLOROWANIA WIERZCHOŁKÓW GRAFU

Problem kolorowania wierzchołków grafu jest jednym z najbardziej znanych problemów NP-zupełnych [7-9,13,14,19-22]. Wiele zagadnień, zarówno teoretycznych, jak wynikających z praktyki, daje się sprowadzić do szeroko pojętego kolorowania grafów. Dzieje się tak zawsze wtedy, gdy spotykamy się z zadaniem podziału zbioru, zawierającego wewnętrzne konflikty między elementami, na podzbiory bezkonfliktowe.

Pokolorowaniem wierzchołkowym grafu $G=(V,E)$ nazywamy taką funkcję $c: V \rightarrow N$, że $\{u, v\} \in E \Rightarrow c(u) \neq c(v)$, tj. dowolne dwa sąsiednie wierzchołki $u, v \in V$ otrzymują różne kolory. Funkcję c będziemy nazywać funkcją kolorującą. Graf G, dla którego istnieje pokolorowanie wierzchołkowe używające k kolorów, nazywamy k-kolorowalnym, a takie pokolorowanie k-pokolorowaniem. Funkcja c generuje w takim przypadku podział grafu G na podzbiory $V_1, V_2, \ldots V_k$ takie, że $V_i \cap V_j = \emptyset$, dla $i \neq j$, i $V_1 \cup V_2 \cup \ldots \cup V_k = V$ [14].

W ostatnich latach sformułowano wiele wariantów problemu kolorowania grafów [14] oraz skonstruowano wiele algorytmów kolorowania ze szczególnym uwzględnieniem metaheurystyk takich jak algorytmy genetyczne [3-6,10,12,16,18]. Ze względu na swą trudność problem GCP był tematem DIMACS Challenge w roku 1993 oraz tematem sympozjum obliczeniowego [9,20-22]. Uzyskane dotąd wyniki nie przesądzają o zdecydowanej wyższości żadnej ze stosowanych metod. Wynika stąd potrzeba kontynuowania badań przy użyciu narzędzi, które dotychczas nie były wykorzystywane, takich jak równoległe algorytmy genetyczne.

3. RÓWNOLEGŁE ALGORYTMY GENETYCZNE

Poniżej przedstawiono dwa zasadniczo różniące się między sobą modele równoległego algorytmu genetycznego, na których oparty został program *PGA_for_GCP*.

3.1. Model z dystrybucją wartości przystosowania (master-slave).

Głównymi cechami modelu z dystrybucją wartości przystosowania jest występowanie jednej populacji oraz scentralizowanej selekcji.

Do implementacji takiego algorytmu zastosowano strukturę, gdzie procesor master przechowuje bieżącą populację rozdzielając jej części składowe do przetwarzania równoległego przez procesory slave. Zrównoleglenie obliczeń dotyczy funkcji przystosowania, krzyżowania oraz mutacji. Po zakończeniu obliczeń procesory slave zwracają nowe osob-

niki wraz z wartościami ich funkcji przystosowania do procesora master, który dokonuje selekcji nowej populacji bieżącej, po czym proces powtarza się.

Model synchronicznego algorytmu master-slave jest relatywnie łatwy do symulacji w systemie jednoprocesorowym. Przy konstruowaniu tego typu algorytmu należy brać pod uwagę relację pomiędzy liczbą procesorów i czasem ich komunikacji a przyśpieszeniem uzyskanym dzięki zrównolegleniu obliczeń.

3.2. Statyczne podpopulacje z migracją (model wyspowy).

Model migracyjny jest najbardziej popularną klasą PGA. Populacja globalna jest podzielona na odseparowane od siebie statyczne podpopulacje, w których przebiegają lokalne procesy ewolucji. Wprowadzony jest dodatkowy operator – *migracja*. Polega ona na okresowym kopiowaniu części osobników danej podpopulacji (najlepszych lub losowych) do innej podpopulacji. Jeśli osobnik może migrować do dowolnej podpopulacji, model nazywa się *modelem wyspowym* [17].

Proces migracji może być kontrolowany przez następujące parametry:

- *topologię*, która definiuje połączenia między wyspami,
- *rozmiar migracji*, określa jak wiele osobników podlega migracji,
- *schemat migracji*, określa które osobniki migrują do innej podpopulacji (najlepsze, najgorsze, losowe) oraz jakie są usuwane (zwykle najgorsze lub losowe),
- *odstęp pomiędzy migracjami* – mierzony w generacjach.

Z punktu widzenia implementacji model taki jest prostym rozszerzeniem sekwencyjnego algorytmu genetycznego, a liczba wysp w najprostszym przypadku odpowiada liczbie procesorów. Koewoluujące populacje składają się z osobników tego samego typu. Na wszystkich wyspach obowiązuje ta sama funkcja przystosowania.

4. DEFINICJE FUNKCJI KOSZTU I FUNKCJI PRZYSTOSOWANIA

Kryterium określającym jakość rozwiązania jest funkcja kosztu (zwana niekiedy funkcją celu). W algorytmie użyto funkcji zliczającej liczbę krawędzi grafu z jednakowo pokolorowanymi końcami z uwzględnieniem całkowitej liczby użytych barw.

Funkcja kosztu dla k-pokolorowania p jest zdefiniowana następująco:

$$f(p) = \sum_{\{u,v\}\in E} q(u,v) + k + d\,,$$

gdzie: $q(u, v) = 2$, gdy $c(u) = c(v)$, w przeciwnym razie $q(u,v) = 0$; $d = 1$, gdy $\sum_{\{u,v\}\in E} q(u,v) > 0$, lub d=0, gdy $\sum_{\{u,v\}\in E} q(u,v) = 0$; k – jest liczbą użytych kolorów.

Tak skonstruowana funkcja kosztu $f(p)$ została wyznaczone empirycznie.

W naszych PGA stosowana jest selekcja proporcjonalna z funkcją przystosowania równą $1/f(p)$.

5. BADANIA SYMULACYJNE

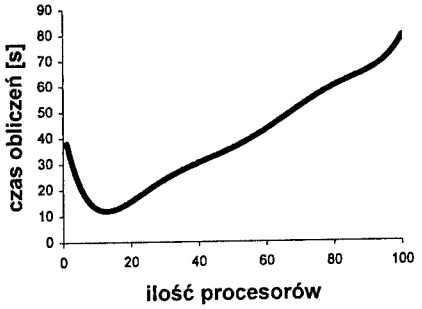
Celem przeprowadzonych eksperymentów było zbadanie drogą symulacji komputerowej wybranych aspektów równoległych algorytmów genetycznych dla problemu GCP i sformułowanie wynikających z badań wniosków.

Wszystkie doświadczenia przeprowadzono na komputerze z procesorem AMD Athlon™ 1700+ (1472 MHz) i pamięcią RAM 256 MB.

5.1. Zależność czasu obliczeń algorytmu od ilości procesorów w modelu master-slave

Parametrem zmiennym w przedstawionych badaniach jest liczba procesorów, na które dzielona jest populacja globalna. Równolegle wykonywane są operacje oceny, krzyżowania i mutacji osobników. Założono, że wszystkie procesory działają z identyczną prędkością, a czas działania programu równoległego jest sumą dwóch składników: czasu działania programu sekwencyjnego podzielonego przez ilość komputerów i czasu komunikacji.

Oczywistym jest fakt wzrostu czasu komunikacji wraz ze zwiększającą się liczbą procesorów. Dla określonego zadania, istnieje pewna optymalna liczba procesorów, dla której czas obliczeń w systemie jest minimalny (rys. 1). Dla grafów większych, gdzie obliczenia funkcji kosztu oraz operacje krzyżowania i mutacji są bardziej czasochłonne, efektywne staje się bardziej drobnoziarniste zrównoleglenie.



Rys. 1. Wpływ ilości procesorów na czas obliczeń w modelu master-slave.

5.2. Wpływ schematu migracji na wyniki obliczeń w modelu wyspowym

Badano wpływ schematu migracji na efektywność algorytmu, mierzoną liczbą generacji n, potrzebnych dla znalezienia liczby chromatycznej grafu. Doświadczenia przeprowadzono z pięcioma grafami z archiwum DIMACS. Każdy pomiar powtórzono 30 razy. Parametry algorytmu: liczebność populacji 60, krzyżowanie SPPX, mutacja First Fit z prawdopodobieństwem 0.1, początkowa ilość kolorów 4, ilość wysp 5, odstęp pomiędzy migracjami 5. Wyniki badań przedstawiono w tablicy 5.1.

Dla wszystkich grafów migracja najlepiej przystosowanych osobników daje najlepsze rezultaty. Ilość generacji n przy zastosowaniu tego rodzaju migracji jest najmniejsza. Inaczej jest z zastosowaniem wymiany losowo wybranych osobników. Tutaj wyniki niewiele się różnią od rezultatów uzyskanych bez zastosowania operatora migracji. Jedynie w grafie *Mulsol.i.4* migracja poprawia znacznie działanie algorytmu. Graf ten posiada

Tablica 5.1

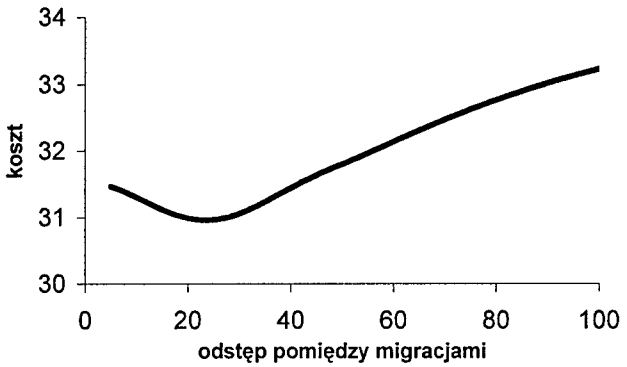
Wpływ schematu migracji na efektywność PGA w modelu wyspowym.

Graf	Wierzchołki	Krawędzie	Kolory	Schemat migracji		
				najlepsze	losowe	bez migracji
				n	n	n
David	87	406	11	77	121	114
Games120	120	638	9	41	64	72
Myciel7	191	2360	8	54	94	86
Queen5	25	160	5	46	78	75
Mulsol.i.4	185	3946	31	1004	1307	2680

posiada dużą liczbę wierzchołków, a co za tym idzie, istnieje większa przestrzeń poszukiwania rozwiązania. Wynika stąd, że dla dużych grafów, także wymiana losowych osobników jest skuteczna.

5.3. Wpływ odstępu pomiędzy migracjami na wyniki obliczeń w modelu wyspowym

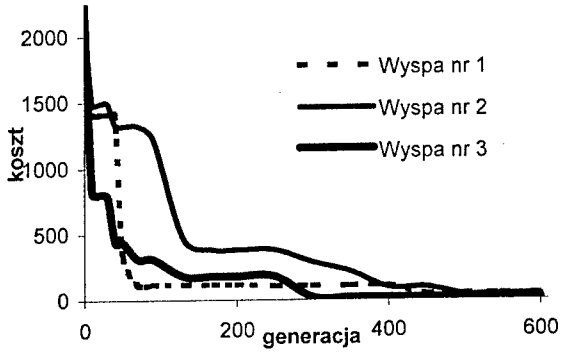
Zbadano także wpływ odstępu między migracjami na jakość uzyskanego rozwiązania mierzoną wartością kosztu najlepszego rozwiązania w modelu wyspowym. Symulacje przeprowadzano na losowo wygenerowanym grafie (75 wierzchołków, 851 krawędzi) przy następujących parametrach: liczebność populacji 60, krzyżowanie SPPX, mutacja porządkowa z prawdopodobieństwem 0.1, początkowa ilość kolorów 10, ilość wysp 3, migracja najlepszych 5 osobników. Parametrem zmiennym jest odstęp pomiędzy migracjami, a koszt oznacza średnią wartość z 30 pomiarów najmniejszego kosztu na wszystkich wyspach, po upływie 1000 generacji.



Rys. 2. Wpływ odstępu pomiędzy migracjami koszt najlepszego rozwiązania w modelu wyspowym.

Napływające osobniki powinny poprawiać zbieżność algorytmu w kierunku najlepszego rozwiązania. Można zauważyć, że istnieje pewna wartość odstępu pomiędzy migracjami, dla której algorytm działa najlepiej (rys. 2). Zarówno poniżej jak i powyżej tej wartości wyniki są słabsze. Zbyt rzadko występująca migracja nie „podciąga" gorszych

wysp. Z kolei zbyt częsta wymiana osobników pomiędzy wyspami również daje gorsze rezultaty. Tłumaczyć to można w następujący sposób: intensywna wymiana nie pozwala na rozwinięcie się na wyspie populacji odmiennej od innych. Dobre osobniki na wszystkich koewoluujących wyspach są do siebie podobne, ponieważ imigrujące osobniki mają duży wpływ na wynik ewolucji na danej wyspie.



Rys. 3. Koszt najlepszego rozwiązania na izolowanych wyspach.

Rola migracji zmienia się na różnych etapach działania algorytmu. Na początku, kiedy na izolowanych wyspach mogą występować znaczne różnice w przystosowaniu najlepszych osobników (rys. 3), migracja wpływa korzystnie na ewolucję na najgorszych wyspach: „zaawansowane" wyspy przekazują swoje osiągnięcia "zacofanym". Na końcowym etapie działania algorytmu, kiedy osobniki na wszystkich wyspach oscylują wokół rozwiązania optymalnego, a wartości ich przystosowania są bardzo zbliżone, celem migracji jest dostarczanie odmiennych, niekoniecznie lepszych, osobników, które w krzyżowaniu z osobnikami z obcej wyspy utworzą poszukiwane najlepsze rozwiązanie. Na tym etapie rzadko występująca migracja wydaje się być bardziej pożądana.

5.4. Porównanie efektywności programów CGA i PGA (model wyspowy)

W doświadczeniu porównano efektywność programów *PGA_for_GCP* (model wyspowy) i CGA (*Constructive Genetic Algorithm*) [5,16] mierzoną liczbą generacji n, potrzebnych dla znalezienia optymalnego pokolorowania i czasem t obliczeń w systemie równoległym. Każdy pomiar powtórzono 30 razy. Parametry algorytmu: liczebność populacji 60, mutacja First Fit z prawdopodobieństwem 0.1, początkowa ilość kolorów 4, ilość wysp 3, odstęp pomiędzy migracjami 5, rozmiar migracji 5. Doświadczenie to przeprowadzono na grupie sześciu grafów o różnych rozmiarach. Wyniki badań zebrane zostały w tablicy 5.2.

Biorąc pod uwagę czas działania do momentu znalezienia optymalnego pokolorowania najlepszym krzyżowaniem jest CEX. SPPX jest krzyżowaniem najmniej skomplikowanym, dlatego czas jego działania jest najkrótszy – wypada on jednak gorzej w porównaniu do CEX, ponieważ algorytm musi działać przez więcej generacji, zanim znajdzie optymalne rozwiązanie. Program *PGA_for_GCP* dzięki nowym operatorom jest zdecydowanie szybszy niż referencyjny program *CGA*.

Porównanie operatorów SPPX i CEX z operatorami UISX [4] i GPX [6] można znaleźć w pracy [11].

Tablica 5.2

Porównanie efektywności dwóch programów genetycznych dla GCP

Graf	Wierz-chołki	Krawędzie	Kolory	Program CGA [5,16]		Program PGA_for_GCP (model wyspowy)			
						Krzyżowanie SPPX		Krzyżowanie CEX	
				n	t [s]	n	t [s]	n	t [s]
Anna	138	493	11	46	108	61	3,2	15	1,3
David	87	406	11	62	44	74	3,6	20	1,0
Huck	74	301	11	12	5	29	0,5	12	0,7
Miles500	128	1170	20	322	2610	152	38	100	3,0
Myciel7	191	2360	8	295	4433	76	7,6	20	1,0
Mulsol_1	197	325	49	42	114	180	34	58	2,0

6. ZAKOŃCZENIE

Przeprowadzając badania przy użyciu programu *PGA_for_GCP*, stwierdzono przyspieszenie obliczeń w modelu z dystrybucją wartości przystosowania, jak również przebadano wpływ liczby procesorów na czas obliczeń w systemie. W modelu z wieloma podpopulacjami istnieje mniejsze prawdopodobieństwo zatrzymania się ewolucji w lokalnym maksimum (w przypadku poszukiwania maksimum funkcji przystosowania), a migracja najlepszych osobników znacznie skraca czas znalezienia optymalnego rozwiązania. Program *PGA_for_GCP* z nowymi operatorami krzyżowania (model wyspowy) jest szybszy od programu *CGA* opisanego w literaturze.

Odpowiednia musi być skala rozwiązywanego problemu. Stosowanie równoległych algorytmów genetycznych dla zbyt łatwych zadań nie odnosi zamierzonych skutków. Po pierwsze zwykłe algorytmy sekwencyjne mogą radzić sobie równie dobrze z takim problemem, po drugie niewspółmierny do korzyści jest nakład pracy, związany z implementacją równoległego modelu.

BIBLIOGRAFIA

[1] Alba E., Tomasini M.: *Parallelism and evolutionary algorithms*, IEEE Trans. Evol. Computing 6 (2002) No.5, s. 443-462.

[2] Bäck T.: *Evolutionary Algorithms in Theory and Practice*, Oxford University Press (1996)

[3] Croitoriu C., Luchian H., Gheorghies O., Apetrei A.: *A new genetic graph coloring heuristic*, Computational Symposium on Graph Coloring and Generalizations COLOR'02, w: Proc. Int. Conf. Constraint Programming CP'02 (2002)

[4] Dorne R., Hao J. -K.: *A new genetic local search algorithm for graph coloring*. Int. Conf. Parallel Problem Solving from Nature PPSN'1998, w: Lecture Notes in Computer Science 1498 s. 745-754., Springer-Verlag 1998.

[5] Filho G.R., Lorena L.A.N.: *Constructive genetic algorithm and column generation: an application to graph coloring*, Proc. Asia Pacific Operarions Research Symposium APORS'2000.

[6] Galinier Philippe, Hao Jin-Kao.: *Hybrid evolutionary algorithms for graph coloring*. Journal of Combinatorial Optimization 3, s. 379–397, Kluwer Academic Publishers 1999.

[7] Garey M. R., Johnson D.S.: *Computers and Intractability. A Guide to the Ttheory of NP-Completeness*. W.H. Freeman and Co., San Francisco 1979.

[8] Jensen T.R., Toft B.: *Graph Coloring Problems*, Wiley Interscience 1995.
[9] Johnson D.S., Trick M.A.: *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Series in Discr. Math. and Theor. Comp. Sc. Vol.26,1996.
[10] Khuri S., Walters T. Sugono Y.: *Grouping genetic algorithm for coloring edges of graph*, Proc. 2000 ACM Symposium on Applied Computing (2000) s. 422--427
[11] Kokosiński Z., Kołodziej M., Kwarciany K.: *Parallel genetic algorithm for graph coloring problem*, Int. Conference on Computational Science ICCS'2004, Kraków, Poland, w: Lecture Notes in Computer Science, Springer-Verlag 2004 (w druku)
[12] Kołodziej M.: *Algorytmy genetyczne dla problemu kolorowania krawędzi grafu*. Praca magisterska, Politechnika Krakowska, Kraków 2003.
[13] Kubale M.: *Introduction to Computational Complexity and Algorithmic Graph Coloring*, GTN, Gdańsk 1998.
[14] Kubale M. (red.): *Optymalizacja dyskretna. Modele i metody kolorowania grafów*, WNT, Warszawa 2002.
[15] Kwarciany K.: *Równoległe algorytmy genetyczne*. Praca magisterska, Politechnika Krakowska, Kraków 2003.
[16] Lorena L.A.N., Filho G.R.: *Constructive genetic algorithm for graph coloring*, Proc. Asia Pacific Operarions Research Symposium APORS'1997.
[17] Nowostawski M., Poli R.: *Parallel Genetic Algorithm Taxonomy*. Proc. Third Intelligent Conference on Knowledge – based Inteligent Information Engineering Systems KES'99, Adelaide, Australia 1999.
[18] Szyfelbein D.: *Metaheuristyki w kolorowaniu grafów*, w: Kubale M. (red.): Optymalizacja dyskretna. Modele i metody kolorowania grafów, s. 26-52, WNT, Warszawa 2002.
[19] de Werra D.: *Heuristics for graph coloring*, w: Tinhofer G. et all. (eds.): Computational graph theory, Springer-Verlag (1990) s. 191—208.
[20] http://mat.gsia.cmu.edu/COLOR/instances.html
[21] ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/
[22] http://mat.gsia.cmu.edu/COLORING03/

PARALLEL GENETIC ALGORITHM APPROACH TO GRAPH COLORING PROBLEM

Summary

In this paper we present results of our experiments with parallel genetic algorithms (PGAs) for graph coloring problem (GCP) that belongs to the class of NP-hard combinatorial optimizations problems. The PGAs' approach is based on co-evolution of a number of populations that exchange genetic information during the evolution process according to a communication pattern. We developed an original computer program *PGA_for_GCP*, which enables selection of various evolution models and its parameters. In computer simulations reported in this paper we used two distinct models, namely the master-slave model and the migration model. In the first model we researched the efficiency of parallel model as a function of the number of slave processors. In the second model we investigated the influence of a migration scheme and a migration distance on the quality of the obtained solutions. We analyzed also the case of evolution on isolated islands. In our experiments we used DIMACS benchmark graphs and two new recombination operators for coloring chromosomes: SPPX (Sum-Product Partition Crossover) in which simple set operations and random mechanisms are implemented, and CEX (Conflict Elimination Crossover) that is focused on the offspring quality. The obtained results are promising and encourage future research focused on PGAs and new genetic operators for graph coloring problems.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Michał Małafiejski, Adrian Kosowski, Marcin Fuszara

Katedra Algorytmów i Modelowania Systemów, Politechnika Gdańska

KONCEPCJA ŚRODOWISKA UMOŻLIWIAJĄCEGO SYMULACJĘ DZIAŁAŃ OBIEKTÓW MILITARNYCH

Streszczenie

Praca zawiera opis najbardziej popularnych platform symulacji walki robotów militarnych. Przeanalizowano ich główne wady i ograniczenia programowe oraz nieadekwatność odtworzenia rzeczywistych warunków i przebiegu walki. Na tym tle przedstawiono oryginalną koncepcję oraz założenia projektowe środowiska symulacji działań autonomicznych obiektów (robotów) militarnych. Proponowana platforma ma umożliwić:

- porównanie jakości taktyk algorytmicznych proponowanych przez różnych programistów (graczy) w warunkach rozgrywanej bitwy,
- kształcenie graczy w zakresie tworzenia wydajnych algorytmów bojowych opartych na sztucznej inteligencji,
- zaprezentowanie programistom idei obiektowego świata wirtualnego, złożonego z autonomicznych jednostek.

1. WPROWADZENIE

Człowiek przez całe tysiąclecia przejawiał zamiłowanie do gier i symulacji militarnych. Na podstawie wzmianek historyków można przypuszczać, że najwcześniejsze pochodzą sprzed ponad 5000 lat. Bez wątpienia ich popularność wynika z kilku różnych czynników. Z jednej strony, prowadzenie symulacji walki pozwala zmierzyć siły z przeciwnikiem, co można interpretować jako przeniesienie idei bezpośrednich starć pomiędzy przeciwnikami na płaszczyznę intelektualną. Z drugiej strony, rozgrywki o charakterze symulacyjnym stanowią rozrywkę dla graczy, jednocześnie pozwalając na trenowanie ich zdolności myślenia taktycznego i wykształcenia odpowiednich nawyków i przyzwyczajeń, potrzebnych dowódcy na polu walki. Następujące z czasem formalizowanie i ujednolicenie zasad rozgrywek pozwoliło na wykształcenie się nowej dziedziny nauki – teorii gier, zajmującej się analizą właściwości poszczególnych gier oraz strategii i podejść zwiększających szanse na zwycięstwo. Ta fascynacja analizą samej istoty gry sprawiła, że z czasem wiodące gry militarne utraciły swój pierwotnie symulacyjny charakter. Podczas, gdy istota gry *Go* znanej w Japonii i Chinach już za czasów Konfucjusza (600 r. p.n.e., [1]) ma pewien związek z zasadami rozstawiania jednostek na polu walki przez starożytnych generałów, to praktykowana w Starożytnym Rzymie gra w warcaby stanowi już skrajne

uproszczenie idei walki zbrojnej w szyku, a datowana na VI w. n.e., wywodząca się z Dalekiego Wschodu, gra w szachy zrywa całkowicie z realiami walki i jej figury mają już znaczenie wyłącznie symboliczne ([2]).

Opisany kierunek rozwoju gier militarnych wynika w dużej mierze z natury działania umysłu ludzkiego, który stara się wyodrębnić abstrakcyjny problem gry od realiów bojowych, co ułatwia analizę sytuacji i rozważania nad dalszymi posunięciami. Współczesna informatyka daje jednak możliwość tworzenia platform symulacji działań wojennych o niespotykanym dotąd realizmie, które mimo to rządzą się czytelnymi prawami i pozwalają na tworzenie bardzo zaawansowanych algorytmicznych taktyk bojowych. Wydaje się naturalne, że postępującej automatyzacji narzędzi używanych w walce (np. sprzętu wspomagającego komunikację, ruch oraz walkę pojazdów) powinno towarzyszyć uniezależnienie procesu sterowania walką od decyzji podejmowanych *ad hoc* przez kontrolującego walkę człowieka. Również w przypadku platform symulacji bojowej, coraz większym uznaniem cieszą się rozwiązania, w których uczestnik (zwany umownie *graczem*) nie steruje bezpośrednio przebiegiem walki, lecz jest ona kontrolowana przez program, realizujący stworzony przez gracza algorytm z opisem taktyki.

W dalszych rozdziałach opisana jest koncepcję platformy symulacji walki obiektów militarnych sterowanych przez programy graczy, która ma w zamierzeniu umożliwić:

- porównanie jakości taktyk algorytmicznych proponowanych przez różnych graczy w warunkach rozgrywanej bitwy,
- opracowywanie nowych podejść do prowadzenia bitwy przy pomocy sztucznej inteligencji i porównanie ich skuteczności ze znanymi zasadami postępowania,
- kształcenie graczy w zakresie tworzenia wydajnych algorytmów bojowych opartych na sztucznej inteligencji,
- zaprezentowanie programistom idei obiektowego świata wirtualnego, złożonego z autonomicznych jednostek (agentów).

W rozdziale 2 przedstawiona jest krótka analiza pewnych istniejących rozwiązań platform symulacji walki, pokazująca, że nie realizują one w pełni wyszczególnionych powyżej celów. W rozdziale 3 podano (w formie postulatów) najważniejsze założenia, które zdaniem autorów musi spełniać platforma symulacji realizująca postawione cele. Propozycję i szkic projektu rozwiązania umożliwiającego symulację walk oddziałów pancernych (sterowanych algorytmami graczy) zawarto w rozdziale 4.

2. OPIS WYBRANYCH PLATFORM SYMULACJI DZIAŁAŃ BOJOWYCH

Przed omówieniem koncepcji planowanego rozwiązania, należy scharakteryzować wcześniejsze popularne symulatory walk autonomicznych jednostek bojowych. Nie uwzględniamy przy tym symulatorów o znaczeniu czysto wojskowym, ponieważ są one zazwyczaj adresowane do wąskiego grona odbiorców i nie odgrywają (obecnie) większego znaczenia z punktu widzenia nauki i rozwoju algorytmów sztucznej inteligencji.

2.1. Wczesne środowiska symulacji walk programów

Prawdopodobnie najbardziej znanym przykładem klasycznego środowiska pozwalającego na przeprowadzenie starcia pomiędzy programami jest maszyna wirtualna MARS, służąca do uruchamiania programów w języku asemblerowym *Redcode* (wykorzystywana

m.in. w turniejach z cyklu *Core War*). Zasady systemu walki opracowano już na początku lat 60-tych na podstawie pomysłu W. Wysockiego, R. Morrisa i D. Ritchiego z Bell Labs [3]. Celem programów uczestniczących w rozgrywce jest zniszczenie przeciwnika poprzez zmianę zawartości pamięci operacyjnej, w której przechowywane są wykonywane przez niego instrukcje. Rozgrywka ma więc charakter bezpośredniego starcia wojennego pomiędzy programami, pozwalając na prowadzenie bitwy mającej niewiele wspólnego z realiami panującymi w prawdziwym świecie. Największą popularnością walki *Core War* cieszyły się w połowie lat 80-tych i od tego czasu trwały intensywne prace nad rozwojem maszyny wirtualnej środowiska, by umożliwić nowe, bardziej wyszukane mechanizmy walki. Wprowadzane modyfikacje dotyczyły m.in. zasad opóźniania reakcji przeciwnika, czy zadawania obrażeń. W efekcie rozbudowa maszyny wirtualnej sprawiła, że rozgrywka przyjęła charakter bardziej przypominający symulację walki robotów, niż bezpośrednie starcie. Wzrost mocy obliczeniowej procesorów i ilości dostępnej pamięci oraz towarzysząca tym zjawiskom popularyzacja wysokopoziomowego programowania obiektowego doprowadziły do całkowitego zerwania z zasadą walk na poziomie języka asemblerowego, zastępując programy walczące przez programy o charakterze obiektów sterujących zachowaniem robotów bojowych działających na specjalizowanych maszynach wirtualnych.

2.2. Platforma Robocode

Platforma programowa Robocode, opracowana w 2000 r. przez konsorcjum IBM, pozwala na symulację walk autonomicznych robotów militarnych. Robotami są czołgi, dla których użytkownicy tworzą w języku Java oprogramowanie sterujące logiką. Roboty umieszczane są na prostokątnej dwuwymiarowej arenie i walczą aż do wyłonienia zwycięzcy lub zniszczenia wszystkich robotów. Środowisko programowe zrealizowane jest w całości w języku Java, a roboty implementowane są poprzez stworzenie klasy dziedziczącej z klasy bazowej robota. Platforma Robocode powstała jako gra dla programistów, lecz szybko dostrzeżono jej walory edukacyjne jako pomoc w nauczaniu języka Java. Obecnie pasja programowania robotów skupia tysiące programistów testujących i dopracowujących zaawansowane algorytmy sztucznej inteligencji, które pozwolą ich robotom pokonać przeciwników i piąć się na szczyty rankingów. Dokładniejszą charakterystykę środowiska Robocode znaleźć można w [4].

Środowisko Robocode, pomimo znacznej popularności, nie jest wolne od wad. Duży nacisk położono na prostotę zasad rządzących środowiskiem, jednakże obniżono przy tym znacznie realizm symulacji oraz okrojono interfejs dostępny dla programisty piszącego robota. Poniżej wymieniono główne czynniki sprawiające, że platforma Robocode nie pozwala w pełni osiągnąć opisanych w rozdziale 1 celów:

- Starcia pomiędzy robotami prowadzone są w systemie klatek czasowych, zbliżonym do systemu stosowanego w klasycznej grze turowej z ograniczeniem czasu. Czas walki podzielony jest na klatki ustalonej długości (zazwyczaj ok. 0.05s), w trakcie których algorytm robota ma czas na podjęcie decyzji o zachowaniu robota w kolejnej klatce. Takie podejście znacząco zmniejsza realizm symulacji i stwarza problemy przy implementacji robota, zmuszając programistę do programowania w sposób nieobiektowy (robot musi działać w sposób sekwencyjny). Jednocześnie, rozwiązanie to w pewien sposób faworyzuje algorytmy wolniejsze, które wykorzystują prawie cały dostępny czas myślenia (nie jest „premiowana" szybsza reakcja; nie ma możliwości podjęcia działania natychmiast po zakończeniu opracowywania ruchu przez algorytm). Nasilająca się tendencja tworzenia robotów wykorzystujących prawie cały

dostępny czas myślenia wywołała efekt uboczny uzależnienia jakości działania robota od szybkości platformy, na której jest przeprowadzana walka. Takie zjawisko można uznać za dopuszczalne w przypadku gier czysto strategicznych (np. w programach do gry w szachy), ale w przypadku symulatora działań militarnych jest bez wątpienia silnie niepożądane.

- Środowisko Robocode nie udostępnia wysokopoziomowych operacji pozwalających na sterowanie ruchem i uzbrojeniem robota. Przykładowo nie ma metod służących do oddawania strzałów z działa w określony punkt na mapie (konieczne jest ręczne sterowanie obrotem wieżyczki), do przemieszczania robota według podanej trasy (nawet łamanej złożonej z odcinków) i do śledzenia ruchu wybranego przeciwnika przy pomocy wiązki promieni radaru. Programista tworzący robota jest zazwyczaj zmuszony samodzielnie zaimplementować te metody, pamiętając o zasadach dyskretyzacji czasu (podział na klatki). Z doświadczeń autorów (opisanych w [5]) wynika, że czasami sterowanie najprostszymi ruchami robota pochłania więcej uwagi programistów niż projektowanie wydajnych algorytmów sztucznej inteligencji.
- Zasady rozgrywek drużynowych w środowisku Robocode wydają się mało dopracowane. Losowe rozmieszczenie robotów na początku starcia, brak korzyści z formowania szyku (ze względu na model lotu pocisków artyleryjskich – powolny ruch pocisku w płaszczyźnie poziomej) i brak opracowanego wysokopoziomowego protokołu pozwalającego na wygodną wymianę informacji i poleceń w obrębie drużyny sprawiają, że zainteresowanie tą odmianą rozgrywki jest najmniejsze, a światowa liga drużyn Robocode liczy niewiele ponad 10 zawodników [6]. Większość twórców robotów zgodnie uważa, że platforma Robocode nie jest w ogóle przystosowana do symulacji zorganizowanej bitwy dwóch drużyn, a tym samym nie pozwala na testowanie i porównywanie algorytmów sztucznej inteligencji związanych ze sterowaniem oddziałami wojska.

Propozycja założeń platformy symulacji obiektów mililtarnych, która jest pozbawiona wymienionych powyżej mankamentów systemu Robocode, została przedstawiona w kolejnym rozdziale.

3. ZAŁOŻENIA KONCEPCYJNE PROPONOWANEGO ROZWIĄZANIA

Dążenie do realizacji postawionych w rozdziale 1. celów wymaga postawienia pewnych założeń dotyczących zasad rządzących wirtualnym światem walczących obiektów militarnych. Prezentowane tu postulaty wynikają z analizy zarówno czynników istotnych z punktu widzenia programisty, jak i koniecznych dla zachowania realizmu symulacji. Wymienione w kolejnych podpunktach założenia są zbieżne ze współczesnymi tendencjami obserwowanymi przy konstrukcji systemów agentowych [7] oraz w rozwoju języków programowania czysto obiektowego, opartego o prototypowanie [8].

3.1. Budowa symulowanego świata w oparciu o jednostki autonomiczne

Większość platform symulacji walki powstających począwszy od połowy lat 90-tych (np. C++ Robots [9] lub omówiony wcześniej Robocode) pozwala na reprezentację jednostki walczącej w postaci klasy implementującej pewien interfejs lub dziedziczącej z pewnej klasy bazowej. Podejście takie, choć w pewnym stopniu obiektowe, zakłada sprzeczne z ideą obiektowości sterowanie robotem przez program zewnętrzny, wywołujący

w odpowiednich sytuacjach ustalone metody robota. Programista traci w tej sytuacji niezależność jego programu od implementacji platformy symulatora (problem ten opisano dla przykładu środowiska Robocode w podpunkcie 2.2).

Proponowana koncepcja symulacji świata opiera się na zasadzie autonomicznej egzystencji wszystkich występujących w nim obiektów (zwanych czasem *jednostkami*). Jednostki dzielą się na *nieożywione* (pasywne), które wykonują pewne zadania zgodnie z ustalonymi regułami, oraz *ożywione* (agentowe), kontrolowane przez sztuczną inteligencję zaprogramowaną przez programistę tworzącego robota lub dostarczaną wraz z platformą symulacji.

Jednostki nieożywione, realizując powierzone im zadania, mogą reagować na bodźce z otoczenia z pewnym opóźnieniem czasowym, zwanym dalej *czasem propagacji*. Może on być uzależniony od warunków panujących w symulowanym świecie, ale nie zależy od sposobu implementacji jednostki (innymi słowy, zakłada się, że jednostki takie myślą z nieskończoną szybkością, a wprowadzane opóźnienia są konieczne dla podniesienia realizmu symulacji). Przykładowo, komunikacja pomiędzy dwoma robotami nie jest natychmiastowa ze względu na opóźnienia wprowadzane umyślnie (w sposób kontrolowany) przez urządzenie nadawcze, powietrze (traktowane jako ośrodek rozchodzenia się fali) i urządzenie odbiorcze.

Sytuacja wygląda zupełnie inaczej w przypadku jednostek ożywionych (agentów). Kluczowym czynnikiem przy symulacji działania agenta jest opóźnienie wynikające z ograniczonej szybkości działania algorytmu sterującego agentem, zwane *czasem myślenia*. Zakłada się, że jednym z zadań systemu sterowania platformą jest taki mechanizm szeregowania zadań i przydziału zasobów sprzętowych, który pozwoli na symulację upływu czasu w taki sposób, aby z punktu widzenia programisty korzystającego z systemu agenci działały (myślały) nieprzerwanie i bezzwłocznie reagowały na bodźce w symulowanym świecie. Jeden z możliwych modeli deterministycznej symulacji upływu czasu spełniający postawione założenie scharakteryzowano w podpunkcie 4.1.

3.2. Oddzielenie symulowanego świata od maszyny wirtualnej systemu sterowania

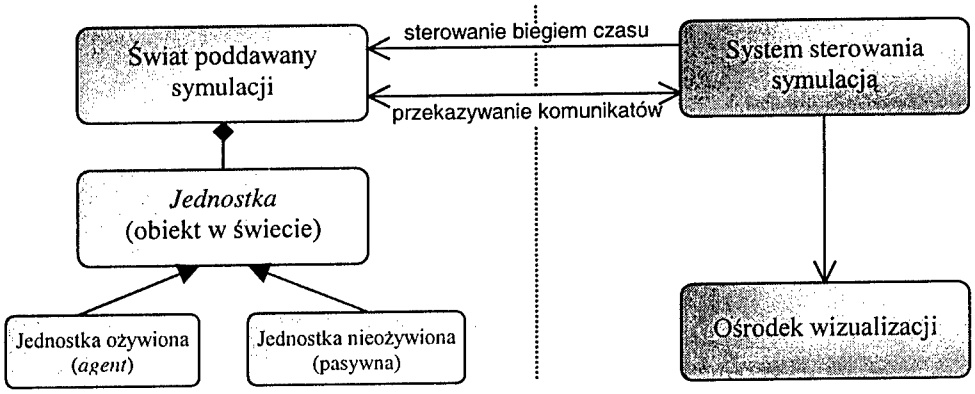
Jednym z podstawowych założeń koncepcyjnych proponowanego środowiska wydaje się wydzielenie dwóch odrębnych podsystemów – systemu sterowania kontrolującego przebieg symulacji oraz systemu opisującego poddawany symulacji świat.

Symulowany świat złożony jest z jednostek nieożywionych oraz agentów, z których część dostarczana jest wraz z środowiskiem symulacji, a część (odpowiedzialna za sztuczną inteligencję wykorzystywaną w walce) tworzona jest przez użytkowników środowiska. Poszczególne jednostki odbierają komunikaty od innych jednostek, co stanowi podstawę dla ich dalszego działania. Wewnętrzna implementacja jednostki jest całkowicie niezależna od zasad działania systemu sterowania. Patrząc z zewnątrz, każda jednostka wydaje się działać na niezależnej maszynie wirtualnej, a zadaniem systemu sterowania jest koordynacja pracy jednostek w celu zachowania ciągłości czasu w symulowanym świecie. W szczególności, prezentowana koncepcja nie narzuca modelu synchronicznego lub asynchronicznego przetwarzania komunikatów, pozostawiając wybór programiście tworzącemu daną jednostkę.

System sterowania można interpretować jako globalną (wspólną dla całego symulowanego świata) maszynę wirtualną. Stanowi ona warstwę pośredniczącą pomiędzy obiektami występującymi w świecie, a zasobami sprzętowymi koniecznymi do przeprowadzenia symulacji i zaprezentowania człowiekowi jej wyników. W obrębie systemu sterowania można wydzielić kilka silnie powiązanych obszarów funkcjonalności:

- funkcjonalność odpowiedzialna za przydział zasobów pamięciowych dla jednostek występujących w symulowanym świecie,
- funkcjonalność pozwalająca na symulację upływu czasu w symulowanym świecie,
- funkcjonalność powiązana z interfejsem użytkownika, pozwalająca na przedstawienie użytkownikowi systemu następujących w czasie zmian symulowanego świata.

Zasadę podziału platformy symulacji na dwie odrębne części przedstawiono schematycznie na rys.1.

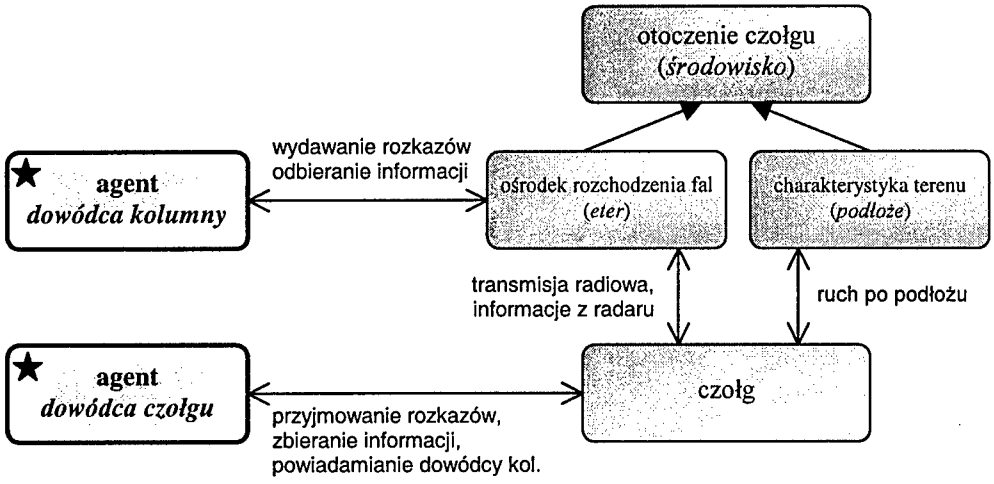


Rys.1. Platforma symulacji jako połączenie symulowanego świata (po lewej stronie linii) i systemu sterującego symulacją (po prawej stronie)

3.3. Skalowalność i modyfikowalność świata

Część spośród jednostek występujących w świecie odgrywa szczególnie ważną rolę, realizując prawa fizyki rządzące wirtualnym światem. Przykładowo, w skład platformy muszą wchodzić jednostki nieożywione odpowiedzialne za realizację komunikacji pomiędzy obiektami (np. „eter" – przestrzeń rozumiana jako nośnik informacji zawartej w fali), czy pozwalające na poruszanie się obiektów wojskowych (np. różne rodzaje podłoża). Platforma dostarcza również opis obiektów militarnych i wyposażenia, którymi bezpośrednio mogą sterować agenci. Przykładowy schemat powiązań logicznych pomiędzy jednostkami prostego świata służącego do symulacji walk czołgów przedstawiono na rys.2.

Wszystkie wymienione elementy mogą być dowolnie rozbudowywane lub dostosowywane do potrzeb konkretnej symulacji, bez konieczności znaczących zmian sztucznej inteligencji sterującej agentami. Użytkownik ma więc istotny wpływ na realizm i dokładność przeprowadzanej symulacji. Jest to istotne ze względów edukacyjnych. Na przykład, tworzenie zaawansowanych algorytmów sztucznej inteligencji można rozpocząć od założenia idealnie płaskiego pozbawionego przeszkód terenu walki, a następnie dostosować tak zaprojektowane algorytmy agentów do walki w terenie górzystym.



Rys.2. Powiązania pomiędzy jednostkami w świecie służącym do symulacji pojedynczej kolumny pojazdów pancernych

3.4. Ujednolicone podejście do procesu programowania sztucznej inteligencji

W podpunktach 3.1 i 3.2 zarysowano zasady działania proponowanego środowiska agentowego do symulacji działań militarnych. Na podstawie przyjętych założeń, można jasno sformułować reguły, które pozwolą programiście na programowanie elementów sztucznej inteligencji wykorzystywanych do sterowania walką.

3.4.1. Taktyka zamiast strategii

Przedstawiony w podpunkcie 3.1 model upływu czasu uwzględnia czas działania algorytmu sterującego agentem jako ważne kryterium, decydujące często o powodzeniu jednostek w walce. Sposób sterowania symulacją jest określony tak, aby z punktu widzenia obserwatora zewnętrznego, agenci działały nieustannie i równocześnie. Takie rozwiązanie zachęca do konstrukcji efektywnych algorytmów sterujących sztuczną inteligencją, a zarazem znacznie podwyższa realizm symulacji (zbliżając sposób „myślenia" agentów do warunków panujących w trakcie rzeczywistego starcia zbrojnego). Oznacza to zerwanie z modelami upływu czasu stosowanymi w strategicznych grach turowych, lub tradycyjnych platformach symulacji, takich jak Robocode.

3.4.2. Sterowanie na wzór człowieka

Projektując algorytmy sztucznej inteligencji dla potrzeb symulacji sterowania obiektami wojskowymi, należy zwrócić uwagę na fakt, że w większości przypadków zadaniem tworzonego algorytmu jest przejęcie roli człowieka, który mógłby wypełniać podobne zadania w rzeczywistym świecie. Prezentowana koncepcja przewiduje zbliżenie zadań agenta do roli człowieka, dając mu możliwość współpracy z innym agentem lub odpowiednią jednostką nieożywioną. Przykładowo, zadaniem agenta – dowódcy kolumny pancernej – może być przesyłanie rozkazów do dowódców poszczególnych czołgów, za pośrednictwem urządzeń transmisyjnych dostępnych w jego pojeździe. Logika dowódcy

jest opracowywana przez programistę korzystającego z platformy symulacji, natomiast oprogramowanie opisujące zasady działania pojazdu dostarczane jest wraz z platformą symulacji.

3.4.3. Hierarchiczna struktura elementów sztucznej inteligencji

Jednym z najbardziej kontrowersyjnych tematów w środowiskach programistów sztucznej inteligencji robotów jest poziom abstrakcji programowania. Niektórzy użytkownicy bez wątpienia preferują programowanie na każdym kroku szczegółowych akcji podejmowanych przez robota, co pozwala na poprawę szybkości i sprawności działania w sytuacjach bojowych. Bez wątpienia jednak platforma symulacyjna, która ma pełnić m.in. rolę programu edukacyjnego oraz środowiska testowego dla zaawansowanych algorytmów dowodzenia oddziałami militarnymi, nie może zmuszać użytkowników do tracenia czasu na oprogramowanie prostych czynności agenta, wykonywanych w prawdziwym świecie przez doświadczonego żołnierza w sposób mechaniczny lub instynktowny.

Proponowane rozwiązanie polega na dostarczeniu wraz z platformą symulacyjną standardowych, szablonowych rozwiązań logiki agenta na wszystkich szczeblach dowodzenia, tzn. przykładowo: agenta-dowódcy sterującego przebiegiem walki ze sztabu, agentów-dowódców wydających rozkazy poszczególnym oddziałom wojska; wreszcie agentów-żołnierzy (sterujących bezpośrednio sprzętem wojskowym). Komunikacja pomiędzy agentami wykorzystuje jednostki nieożywione świata do fizycznej transmisji rozkazów, a każdy z agentów wykonuje (lub ignoruje) rozkazy zgodnie ze specyficzną dla niego logiką.

Użytkownik platformy symulacji może zastąpić lub rozbudować logikę sterującą dowolnie wybranych agentów. Testowanie algorytmów sztucznej inteligencji związanych z dowodzeniem oddziałów wojskowych utrzymanych w odpowiedniej formacji można przeprowadzić poprzez wprowadzenie własnej logiki dla agenta-głównodowodzącego. Użytkownik dysponujący większą ilością czasu może zająć się logiką niżej postawionych oficerów i żołnierzy, co można interpretować jako odpowiednik prowadzenia ćwiczeń i szkoleń wojskowych w rzeczywistym świecie. Pozwala to również uwzględnić wpływ czynników psychologicznych występujących wśród walczących na przebieg walki (dając żołnierzom możliwość odmowy wykonania rozkazu lub sprawiając, że w pewnych sytuacjach rozkazy wykonują mniej lub bardziej skutecznie). Otwiera to nowe możliwości porównywania jakości algorytmów walki (uzależniając taktykę od specyficznego rodzaju prowadzonej armii).

4. ASPEKTY IMPLEMENTACYJNE PROJEKTOWANEJ PLATFORMY

4.1. Realizacja wirtualnej osi czasu w symulowanym świecie

Budowa świata złożonego z jednostek autonomicznych, scharakteryzowanego w podpunktach 3.1 i 3.2, wymaga, aby wszystkie jednostki mogły koegzystować jednocześnie i wchodzić ze sobą w interakcję. Determinizm działania systemu nie jest bezwzględnym wymogiem; bardzo często wręcz sama logika agentów wprowadza pewien czynnik losowy. Jednakże przyjmując, że agenci myślą z ograniczoną szybkością, a czasy propagacji wszystkich jednostek nieożywionych są istotnie dłuższe niż czas wykonywania rozkazów na procesorze realizującym symulację, to prosty model systemu sterowania, obsługujący sekwencyjnie komunikaty pojawiające się na osi czasu, pozwala na przeprowadzenie deterministycznej symulacji świata. Przeprowadzenie symulacji współbieżnej przy wyko-

rzystaniu większej liczby procesorów najłatwiej przeprowadzić metodą predykcji, symulując działanie jednostek na różnych procesorach w sposób niezsynchronizowany, zakładając chwilowy brak interakcji pomiędzy jednostkami. W sytuacji, gdy jedna z jednostek prześle komunikat do jednostki działającej współbieżnie na innym procesorze, to może okazać się konieczny nawrót – wycofanie stanu jednostki-adresata do wcześniejszego stanu tej jednostki, w którym powinna się znajdować w momencie otrzymania przesłanego komunikatu. Eliminowanie tego typu niepożądanych zjawisk poprzez odpowiedni przydział jednostek do procesorów oraz zmniejszenie częstości przesyłania komunikatów przy jednoczesnym zachowaniu realizmu symulacji należą do największych wyzwań implementacyjnych przy realizacji prezentowanej platformy.

4.2. Środowisko implementacji oprogramowania

Przedstawiona koncepcja platformy pozostawia znaczną dowolność w zakresie doboru narzędzi i języków potrzebnych do jej implementacji. Przy podejmowaniu decyzji należy kierować się przede wszystkim możliwościami współbieżnego lub pseudowspółbieżnego działania jednostek autonomicznych. Najbardziej odpowiednie do tego celu są języki o charakterze czysto obiektowym, przystosowane do tworzenia programów opierających się na zasadzie obsługi i generowania zdarzeń, takie jak Java, Smalltalk, a szczególnie języki obiektowe najnowszej generacji z prototypowaniem, np. Io ([10]) lub Self ([11]). Narzucenie ustandaryzowanego protokołu przesyłania komunikatów pomiędzy jednostkami (np. korzystając ze specjalnie zdefiniowanego języka w składni XML) pozwala na tworzenie wybranych jednostek nieożywionych w języku programowania innym, niż reszta platformy symulacyjnej.

Z punktu widzenia końcowego użytkownika systemu największe znaczenie ma język programowania wykorzystywany przy tworzeniu agentów. Wydaje się celowe, by programy agentów nie były wykonywane bezpośrednio, ale pisane w dedykowanym języku interpretowanym przez system sterowania symulacją. Pozwala to uniknąć problemów związanych z pomiarem czasu myślenia agenta, a także poprawia bezpieczeństwo systemu.

5. ZAKOŃCZENIE

Zaprezentowana koncepcja ma charakter dokumentu wstępnego, poprzedzającego prace projektowe oraz implementacyjne, zmierzające do stworzenia środowiska symulacji działań kolumn pancernych czołgów w trójwymiarowym terenie z elementami górzystymi. Dotychczas w ramach Koła Naukowego Sfera, działającego przy Wydziale Elektroniki, Telekomunikacji i Informatyki PG, opracowano język (wraz z interpreterem) przystosowany do programowania logiki agentów oraz protokół komunikacji pomiędzy jednostkami autonomicznymi.

Bardziej szczegółowy opis zastosowanego modelu czasu, efekty dalszych prac nad platformą oraz wyniki prowadzonych przy jej pomocy badań algorytmów sztucznej inteligencji do sterowania oddziałami wojskowymi będą przedmiotem dalszych publikacji.

Przewiduje się dalsze zastosowanie projektowanej platformy do innych celów, np. symulacji działań nie tylko obiektów rywalizujących, ale również współpracujących, a także do badania scenariuszy interakcji pomiędzy obiektami.

BIBLIOGRAFIA

[1] American Go Association: *A Very Brief History of Go.*
http://www.usgo.org/
[2] Masters J.: *Chess – An Online guide.*
http://www.tradgames.org.uk/
[3] Perry J.: *Core Wars Genetics: The Evolution of Predation.*
http://www.corewars.org/
[4] IBM alphaWorks: *Robocode Home Page*
http://robocode.alphaworks.ibm.com/
[5] Fuszara M., Kosowski A.: *Strategie algorytmiczne rywalizacji autonomicznych obiektów militarnych na platformie dedykowanej Robocode.* Referat zgłoszony na II Krajową Konferencję Technologie Informacyjne, Wydział ETI PG. Gdańsk, 2004.
[6] *The Robocode Repository.*
http://www.robocoderepository.com/
[7] Inverno M., Kinny D., Luck M., Wooldridge M.: *A Formal Specification of dMARS.* W: Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages. Lecture Notes in Artificial Intelligence, 1365, 155-176, Springer-Verlag, 1998.
[8] Wikipedia: *Prototype-based programming.*
http://en.wikipedia.org/wiki/Prototype-oriented
[9] Roglnie R.: *C++Robots.*
http://www.gamerz.net/~c++robots/
[10] Dekorte S.: *Io – a small Programming Language.*
http://www.iolanguage.com/
[11] Sun Microsystems: *The Self Programming Language.*
http://research.sun.com/self/language.html

OUTLINE OF AN ENVIRONMENT FOR THE SIMULATION OF AUTONOMOUS MILITARY UNITS

Summary

The paper contains a description of some of the most popular combat simulation environments for military robots. We present their major disadvantages and limitations and further describe the inadequacy of the quality of the adopted simulation process. Using this background information an original concept of an environment for the simulation of autonomous military objects (robots) is presented, and its design assumptions are outlined. The principle aims set before the designed environment require that it should:

- enable the comparison of the quality of algorithmic approaches put forward by different programmers (players) under battle simulation conditions,
- enable the development of the algorithmic skills of the players with respect to military artificial intelligence,
- serve educational purposes, by laying out the principles of an object-oriented world consisting of autonomous units to programmers.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Stefan Pepol, Konrad Piwakowski

Katedra Algorytmów i Modelowania Systemów, Politechnika Gdańska

PAKIET WSPOMAGANIA TWORZENIA I OPTYMALIZACJI HARMONOGRAMÓW ZAJĘĆ AKADEMICKICH

Streszczenie

Przedstawiamy model teoretyczny oraz praktyczne rezultaty wdrożenia pakietu wspomagania tworzenia i optymalizacji harmonogramów zajęć na wydziale ETI Politechniki Gdańskiej. Pakiet ten zawiera implementację algorytmów wyczerpującego przeszukiwania przestrzeni kombinatorycznej oraz przeszukiwania tabu (ang. *Tabu Search*). W modelu obliczeniowym uwzględniono szereg czynników oceny jakości jak np. przerwy w zajęciach, równomierne rozłożenie zajęć, przemieszczanie uczestników pomiędzy salami itp. Oprócz modelu teoretycznego, prezentujemy wyniki eksperymentów obliczeniowych oraz podstawowe informacje o architekturze systemu.

1. WSTĘP

Układanie harmonogramów zajęć szkolnych jest problemem algorytmicznym, w rozwiązywaniu którego, człowiek wciąż pozostaje niedościgniony w konfrontacji z komputerem. Heurystyki stosowane przez ludzi stanowią przedmiot badań, a ich wyniki są często wykorzystywane przez twórców algorytmów formalnych do naśladowania postępowania człowieka. Z drugiej strony problem układania harmonogramów stanowi szczególny przypadek problematyki szeregowania zadań oraz kolorowania grafów. Stąd wielokrotnie spotyka się próby wykorzystania algorytmów opracowanych do rozwiązywania ogólnych problemów teoretycznych z tych dziedzin w oderwaniu od sposobu rozwiązywania tego problemu przez człowieka. W niniejszym artykule prezentujemy efekty wysiłków zmierzających do wykorzystania obu powyższych podejść w celu wytworzenia pakietu oprogramowania wspomagającego pracę człowieka w konkretnym zastosowaniu tj. do układania harmonogramów zajęć akademickich na wydziale ETI Politechniki Gdańskiej.

2. MODEL FORMALNY

Niech P, W, G, Z, S oznaczają odpowiednio skończone zbiory przedmiotów, wykładowców, grup studentów, zajęć oraz sal dydaktycznych. Niech T oznacza rodzinę jedno, dwu i trójelementowych podzbiorów godzin w rozkładzie tygodniowym, przy czym zbiory dwuelementowe (trójelementowe) obejmują wyłącznie pary (trójki) godzin następujących bezpośrednio po sobie. Zbiór zajęć stanowi podzbiór zbioru $P \times W \times G$.

Problem polega na wyznaczeniu harmonogramu tj. różnowartościowej funkcji $f: Z \rightarrow S \times T$, spełniającej zadane warunki. Warunki te dzielimy na krytyczne i niekrytyczne. Harmonogram, w którym jeden z warunków krytycznych jest niespełniony nazywamy sprzecznym. Harmonogram taki nie może być wykorzystany w praktyce. Warunki niekrytyczne są pomocne przy ocenianiu jakości ułożonych harmonogramów. Przykładem warunku krytycznego jest to, że wykładowcy i grupy studentów mogą uczestniczyć w co najwyżej jednych zajęciach w danej godzinie. Natomiast równomierny rozkład zajęć w tygodniu jest przykładem warunku niekrytycznego.

2.1. Przestrzeń przeszukiwań

Przestrzeń przeszukiwań stanowi zbiór $X = (S \times T)^{|Z|}$. Dla elementu $u = (u_1, u_2, ..., u_{|Z|}) \in X$, para $u_i = (u_{si}, u_{ti})$ oznacza salę i zbiór godzin przydzielone i-tym zajęciom. Harmonogram $u \in X$ jest sprzeczny, gdy dla dwóch różnych zajęć i, j posiadających wspólnego uczestnika (grupę studencką lub wykładowcę) lub tę samą salę ($u_{si} = u_{sj}$) zachodzi $u_{ti} \cap u_{tj} \neq \emptyset$. Mówimy wówczas, że elementy u_i i u_j wykluczają się.

2.2. Ocena rozwiązań – funkcja celu

Niesprzeczne harmonogramy różnicowane są pod względem spełnienia warunków niekrytycznych – odpowiadających wielu kryteriom. Aby móc porównać jakość dwóch różnych harmonogramów i w efekcie wybrać lepszy, zachodzi potrzeba zdefiniowania relacji porządku liniowego $\leqq$ w zbiorze X. Relację tę definiujemy wykorzystując funkcję celu $g: X \rightarrow R$, przyjmując $x \leqq y \Leftrightarrow g(x) \leq g(y)$. A zatem oczekujemy, że funkcja g przyjmuje tym większą wartość rzeczywistą im lepszy jest harmonogram. Dobór i konstrukcja funkcji celu jest bardzo istotna, gdyż służy temu, aby algorytm i człowiek dokonywali możliwie jak najbardziej zbliżonej oceny jakości.

Innymi słowy na ocenę jakości harmonogramu zajęć składa się wiele czynników, tworzących osobne kryteria oceny. Żeby mogły one być łatwo uwzględniane w procesie automatycznego planowania, muszą zostać sprecyzowane tak, by oceny dały się wyrazić za pomocą liczb. Jedne z nich mogą być ważniejsze, inne mniej ważne, a wszystkie razem muszą tworzyć, dające się porównać, oceny poszczególnych rozwiązań. W naszym modelu i-temu kryterium oceny przypisywana jest waga $w_i \in R$. Jako końcową ocenę rozwiązania przyjmujemy sumę ważoną $\sum o_i w_i$ ocen o_i przyznanych za spełnienie poszczególnych warunków niekrytycznych.

Aby ustalona wartość wagi, przekładała się w sposób proporcjonalny na udział tego kryterium w końcowej ocenie, wartości ocen za poszczególne kryteria powinny mieścić się w tym samym zakresie, tzn. najlepsza i najgorsza możliwa ocena powinny być takie same dla każdego kryterium oceny. Jednak specyfika poszczególnych czynników oceny oraz sposobu ich obliczania powoduje, że oceny za różne kryteria należą do różnych przedziałów, dlatego dokonywana jest normalizacja ocen, tak by sprowadzać wszystkie obliczone oceny do jednego zakresu. Obliczanie ocen rozwiązań, jak wynika z przeprowadzonych prób, jest czynnością, która ma największy udział czasowy w procesie szukania rozwiązań, a więc przetwarzanie ocen powinno być przeprowadzane możliwie małym kosztem obliczeniowym. Mając to na uwadze, oceny dotyczące spełnienia poszczególnych kryteriów normalizowane są liniowo tak, aby najgorszej możliwej ocenie danego kryterium odpowiadała znormalizowana ocena równa zero, natomiast najlepszej pewna ustalona wartość $c > 0$. Ponieważ ustalenie skrajnych wartości ocen przed normalizacją, jakie mogą wystąpić dla poszczególnych kryteriów może być trudne, liczby te są aproksymowane i tak, dla i-tego kryterium, jako minimalną (maksymalną) możliwą ocenę $x_i^{\min}$ ($x_i^{\max}$) przyjmuje się

najmniejszą (największą) ocenę, jaka wystąpiła dotąd w trakcie wykonywania algorytmu. Zmiana wartości x_i^{min} lub x_i^{max} (wskutek znalezienia nowego najgorszego lub najlepszego harmonogramu) wymaga zatem wykonania ponownej normalizacji ocen zapamiętanych rozwiązań. Niech p_1, p_2, ..., p_k oznaczają nieznormalizowane oceny spełnienia poszczególnych warunków przez harmonogram x. Wartość funkcji celu obliczana jest według następującego wzoru:

$$g(x) = \sum_{i=1}^{k} w_i(a_i p_i + b_i) \tag{2.1}$$

gdzie: $a_i = \dfrac{c}{x_i^{max} - x_i^{min}}$

$$b_i = \frac{x_i^{min} \cdot c}{x_i^{min} - x_i^{max}}$$

Współczynniki a_i i b_i przekształceń liniowych zostały dobrane tak, aby

$$a_i x_i^{min} + b_i = 0 \text{ oraz } a_i x_i^{max} + b_i = c .$$

Kierując się opisaną wyżej funkcją celu, można szukać dobrych rozwiązań tzn. takich, dla których g przyjmuje stosunkowo dużą wartość i je zapamiętywać, a pomijać te, dla których wartość ta jest mała, czyli rozwiązania gorsze.

Poniżej przedstawiamy poszczególne kryteria oraz sposób obliczania dla nich wartości ocen w naszym systemie.

2.3. Godziny zajęć w ciągu dnia lub tygodnia

Jednym z czynników wpływających na ocenę planu jest to, w jakich godzinach odbywają się poszczególne zajęcia. Przy czym ocena ta może być zróżnicowana dla poszczególnych dni, lub jednolita dla wszystkich dni.

Regułą dotyczącą planów realizowanych na wydziale ETI PG jest to, że poza pewnymi wyjątkami, każda godzina zajęć wraz z przerwą obejmuje na grafiku jedną godzinę zegarową, zaczynając od pewnej pełnej godziny i kończąc na następnej. Dzięki temu w tego typu ocenie wystarczy rozróżniać punktację dla poszczególnych godzin zegarowych, w których mogą się odbywać zajęcia. Wszystkie dni tygodnia, dla których układany jest plan zajęć mogą być traktowane jednakowo lub oddzielnie. W szczególności zaproponowano oddzielne traktowanie poniedziałku i piątku. Osobna punktacja dla tych dwóch dni daje m.in. możliwość uzyskiwania planów zajęć, w których studenci nie mają zajęć w poniedziałek rano i w piątek wieczorem, dzięki czemu czas wolny od zajęć, obejmujący weekend jest dłuższy, co jest szczególnie istotne dla studentów na stałe mieszkających w znacznej odległości od uczelni.

Ocena harmonogramu dla tego kryterium jest obliczana jako średnia arytmetyczna ocen wyznaczonych dla każdej grupy osobno. Ocena dla jednej grupy jest wyznaczana jako suma punktów za każdą godzinę, w której odbywają się zajęcia tej grupy.

2.4. Okienka w planach zajęć studentów

W rozkładach zajęć studentów, w ciągu jednego dnia mogą zdarzać się okresy czasu, w których dana grupa nie ma zajęć. Takie przerwy w zajęciach nazywamy *okienkami.* Są one zjawiskiem niepożądanym, ponieważ wydłużają czas od rozpoczęcia do zakończenia zajęć w danym dniu, chociaż mogą one w pewnych przypadkach stanowić czynnik pozytywnie wpływający na ocenę planów zajęć, np. w przypadku jednogodzinnego okien-

ka, rozdzielającego dużą liczbę godzin zajęć, które może stanowić przerwę na odpoczynek lub posiłek.

Podobnie rozumiane okienka w planach zajęć prowadzących nie mają tak dużego znaczenia, ze względu na prace naukowe prowadzone przez pracowników uczelni, które najczęściej mogą być bez przeszkód przeplatane z zajęciami dydaktycznymi.

Analogicznie jak w przypadku poprzedniego kryterium, ocena rozwiązania to średnia arytmetyczna ocen wszystkich grup studentów, dla których są planowane zajęcia, które z kolei są obliczane jako suma punktów za każde okienko w planie zajęć, bez względu na to czy jest to okienko spowodowane wprowadzeniem nowych zajęć czy było już ono zanim rozpoczęto planowanie. Poszczególne okienka są punktowane ze względu na ich rozmiar, czyli liczbę godzin. Mogą być to zarówno punkty ujemne (karne) jak i nieujemne. Przy ich ustalaniu należy pamiętać, że brak okienka można rozumieć jako okienko o zerowej długości, za które przyznawane jest zawsze zero punktów, a więc dodatnie punkty za okienka będą promowały plany zajęć, w których takie przerwy występują, zaś punkty ujemne będą powodowały unikanie okienek.

2.5. Liczba zajęć w ciągu dnia

Ważnym czynnikiem wpływającym na jakość harmonogramów zajęć, jest sposób rozłożenia zajęć w planach studentów, ze względu na liczbę godzin zajęć dziennie. Podstawowym warunkiem, do spełnienia którego chcemy dążyć jest to, aby zajęcia były rozłożone równomiernie pomiędzy dniami tygodnia. Możliwość wyrażenia i spełnienia tych oczekiwań daje ocenianie rozwiązań ze względu na liczbę godzin zajęć dydaktycznych w każdym dniu. Dla każdej wartości tego parametru począwszy od zera do maksymalnej możliwej liczby godzin w jednym dniu, przydzielana jest liczba punktów (która może być także ujemna). Dla każdej grupy studentów, która uczestniczy w planowanych zajęciach, obliczana jest suma punktów za wszystkie dni. Uśredniona wartość obliczonych sum punktów dla grup stanowi ocenę rozwiązania. Przy tak zdefiniowanym sposobie oceny planów poszczególnych grup, założony cel można osiągnąć przez ustalenie punktacji tak, by liczba punktów progresywnie malała wraz ze wzrostem liczby zajęć w dniu. Więcej informacji na temat metod doboru funkcji oceniającej warunek równomiernego rozkładu zajęć w tygodniu przedstawiono w pracach [1] oraz [2].

2.6. Preferowane godziny przez prowadzących

W przyjętym modelu ocena spełnienia warunku zaplanowania zajęć wykładowców w godzinach przez nich preferowanych jest obliczana jako suma wszystkich zajęć, które odbywają się w godzinach zadeklarowanych przez prowadzących je wykładowców jako preferowane.

Niezależnie od godzin preferowanych, wykładowcy określają godziny „zabronione". Warunek nie planowania zajęć w godzinach zabronionych jest traktowany jako krytyczny.

2.7. Budynek, w którym odbywają się zajęcia

Z pewnych względów budynek, do którego należy sala dydaktyczna, w której odbywają się poszczególne zajęcia, jest parametrem wpływającym na jakość harmonogramów zajęć. Najkorzystniejszą jest sytuacja, gdy zajęcia odbywają się w budynku prowadzącego je wydziału. Wynika to chociażby z tego, że administracja każdego wydziału w największym stopniu może dysponować własnymi salami. Ponadto oczywistym jest, że większość zajęć odbywa się w budynku lub budynkach rodzimego wydziału, stąd próba

zaplanowania jak najmniejszej liczby zajęć w innych budynkach może pośrednio powodować minimalizowanie liczby zmian budynku, w jakim odbywają się zajęcia pewnej grupy studentów lub prowadzącego w danym dniu, co jest sytuacją pożądaną. W danych przechowywanych w systemie funkcjonującym na wydziale ETI PG, poza budynkiem wydziałowym, rozróżniane są dwie kategorie: inne budynki należące do uczelni oraz budynki spoza uczelni. Każdej kategorii budynku można przypisać liczbę punktów przysługującą za zajęcia, które odbywają się w budynku należącego do tej kategorii. Ocena spełnienia warunku kategorii budynku jest wyliczana jako suma punktów za kategorię budynku dla wszystkich planowanych zajęć.

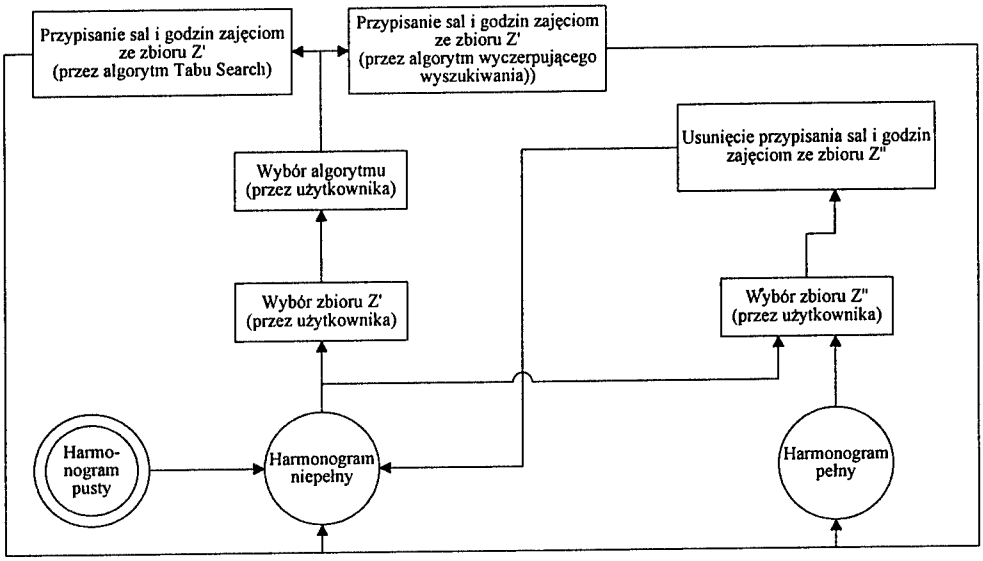
2.8. Status sali, w której odbywają się zajęcia

Z danych systemu dotyczących sal dydaktycznych można odczytać też parametr nazywany *statusem sali*. Decyduje on o tym czy sala jest oznaczona jako *własnościowa* czy *pożyczona*. Te pierwsze uznawane są za korzystniejsze w przypadku planów zajęć, więc każde zajęcia w takiej sali promowane są jednym punktem dodawanym do oceny warunku statusu sali. Sala może mieć też status mówiący, że jest ona niedostępna, wówczas nie jest uwzględniana podczas planowania zajęć.

3. ALGORYTMY

W ramach omawianego pakietu zaimplementowano algorytm wyczerpującego przeszukiwania oraz algorytm Tabu Search (TS). Pierwszy z nich posiada wykładniczą złożoność czasową i może być wykorzystywany do znajdowania optymalnych rozwiązań dla bardzo ograniczonej przestrzeni przeszukiwań – np. do zmiany pełnego niesprzecznego harmonogramu, gdy okazuje się, że jeden z wykładowców deklaruje dodatkowe godziny zabronione i istniejący harmonogram staje się sprzeczny. Wówczas algorytm wyczerpujący może być wykorzystany do znalezienia nowego harmonogramu, który możliwie jak najmniej różni się od dotychczasowego (w takim przypadku bardzo ważne jest, aby jak najmniej zajęć zmieniło salę lub godzinę). Algorytm TS przeznaczony jest do znajdowania niesprzecznych rozwiązań, o możliwie wysokiej wartości funkcji celu, dla dużych przestrzeni przeszukiwań.

Założono interaktywny sposób definiowania zadań dla obu algorytmów. Każda funkcja częściowa $h^* : Z \rightarrow S \times T$ odpowiada harmonogramowi, w którym nie koniecznie wszystkie zajęcia mają przyporządkowane salę i zbiór godzin. Niech $h^{*(-1)}(S \times T) = Z_c$ (zbiór zajęć, które mają już przyporządkowane sale i godziny) oraz $Z_v = Z - Z_c$. Założono, że proces układania harmonogramu przebiega poprzez kolejne kroki uszczegółowiania częściowych harmonogramów. Początkowo zbiór Z_c jest pusty, a $Z_v = Z$. W kolejnych krokach zbiór Z_v zmniejsza się, a zbiór Z_c maleje. Każdy taki krok oznacza wybranie podzbioru zajęć $Z' \subseteq Z_v$ i przypisanie godzin i sal tym zajęciom (poprzez zastosowanie jednego z algorytmów) tak, aby uszczegółowiony harmonogram dla zajęć $Z_c \cup Z'$ był niesprzeczny. Po wykonaniu iteracji zbiór Z_c jest zastępowany przez $Z_c \cup Z'$, a Z_v przez $Z_v - Z'$. Proces zostaje zakończony, gdy $Z_c = Z$ (i $Z_v = \varnothing$). W każdym kroku wybór zbioru Z' i algorytmu dokonywany jest przez użytkownika. Interfejs pakietu umożliwia użytkownikowi wiele różnorodnych metod wyboru zbioru Z' jak np. zbiór zajęć związanych z danym uczestnikiem (wykładowcą lub grupą studentów). Możliwe są także kroki odwrotne, np. gdy użytkownik nie akceptuje przypisań sal i godzin dla pewnego podzbioru $Z'' \subseteq Z_c$ (np. dlatego, że zmienił się zbiór sal, którymi dysponuje wydział). Wówczas zbiór Z_c jest zastępowany zbiorem $Z_c - Z''$, a zbiór Z_v zbiorem $Z_v \cup Z''$. Diagram stanów i zdarzeń opisujący opisany powyżej proces został przedstawiony na rys. 1.



Rys.1. Diagram interakcji układania harmonogramu

Szczegółowe informacje dotyczące implementacji algorytmu wyczerpującego przeszukiwania z nawrotami i przewidywaniem kolizji zamieszczone są w [1] i [3]. Ogólne informacje o paradygmacie przeszukiwania tabu można znaleźć w [4] i [5], a szczegóły implementacyjne tej metody w omawianym pakiecie, a w szczególności organizację struktury sąsiedztwa, definicje funkcji stanu i funkcji aspiracji, sposób wyboru kolejnego stanu, organizację listy tabu, wybór rozwiązania początkowego oraz warunki akceptacji rozwiązania końcowego przedstawiono w [1]. Inne przykłady zastosowania metody Tabu Search do układania harmonogramów zajęć przedstawiono między innymi w [6] i [7].

4. WYNIKI PRAKTYCZNE

Poniżej prezentujemy wyniki praktycznych eksperymentów z wykorzystaniem omawianego pakietu.

4.1. Przeszukiwanie wyczerpujące

Algorytm wyczerpujący zawsze znajduje rozwiązania optymalne. Zasadniczym problemem jest czas potrzebny do znalezienia tego rozwiązania. Głównymi czynnikami wpływającymi na czas obliczeń jest liczba zajęć do zaplanowania $|Z'|$ oraz liczba "wolnych" par ze zbioru $S \times T$.

Przeszukiwanie wyczerpujące ma złożoność o stopniu wykładniczym. Z tego powodu z góry wiadomo było, że nie będzie go można wykorzystywać do planowania zajęć w liczbie większej niż pewna graniczna wartość, po przekroczeniu której, czas przetwarzania będzie gwałtownie rósł. Niewiadomą była sama wspomniana graniczna wartość. Tablica 4.1 przedstawia czasy przeszukiwania rozwiązań w zależności od liczby planowanych zajęć. Jej wiersze odnoszą się do kolejnych przeprowadzanych prób. Wyniki, które przekraczają 10 minut, nie są pomiarem końcowym a jedynie oszacowaniem, tym mniej dokładnym im dłuższy jest przewidywany czas przetwarzania. Szacowany całkowity czas

szukania został obliczony przez program na podstawie czasu, który upłynął od momentu rozpoczęcia procesu oraz liczby sprawdzonych liści drzewa przeszukiwań względem łącznej ich liczby, obliczanych przy założeniu, że wierzchołki odrzucone przez funkcję akceptacji także należą do drzewa.

Jak wynika z przeprowadzonych testów, w typowych przypadkach algorytm wyczerpujący można stosować do planowania zaledwie kilku zajęć jednocześnie.

Tablica 4.1

Czas przeszukiwania wyczerpującego w zależności od liczby zajęć

Eksperyment	Liczba planowanych zajęć					
	1	2	3	4	5	6
1	< 1 sek.	< 1 sek.	1 sek.	1 sek.	16,5 godz.	3:36 min.
2	< 1 sek.	1 sek.	7 sek.	12 sek.	34 godz.	2 dni
3	< 1 sek.	1 sek.	16 sek.	19 min.	43 godz.	2 dni
4	< 1 sek.	1 sek.	55 sek.	52 min.	48 godz.	4 dni
5	< 1 sek.	2 sek.	129 sek.	2:09 min.	17 dni	9 dni
6	< 1 sek.	3 sek.	515 sek.	20:16 min.	64 dni	2 lata
Średnio	**< 1 sek.**	**1,4 sek.**	**121 sek.**	**3:56 min.**	**14 dni**	**125 dni**

Czas szukania rozwiązań dla określonej liczby zajęć może być bardzo zróżnicowany, co wynika z różnej liczby istniejących rozwiązań dla poszczególnych przypadków. W szczególności planowanie zajęć grupy, która ma prawie w całości ułożony plan, będzie trwało znacznie krócej niż w przypadku braku rozplanowanych dla niej zajęć, ponieważ będzie mniej możliwości poprawnego ulokowania nowych zajęć w określonym czasie. Liczbę rozwiązań może dodatkowo ograniczyć mniejsza liczba dozwolonych godzin zadeklarowanych przez wykładowców oraz ograniczona dostępność sal.

Może także dojść do sytuacji, w której planowanie coraz większej liczby zajęć będzie trwało coraz krócej. Wynika to także ze zmniejszenia liczby rozwiązań, powodowanej tym, że „trudniej" jest przydzielić większej liczbie zajęć te same zasoby, tak by uzyskać poprawne harmonogramy.

Prędkość przeszukiwania rozwiązań przez algorytm także zależy od rozmiarów danych wejściowych, chociaż oczywiście w znacznie mniejszym stopniu. Dla przeprowadzonych eksperymentów wynosiła ona kilka milionów rozwiązań na minutę (wraz z oceną rozwiązań). Zależy ona także od charakteru zadania, czyli np. maleje ze wzrostem liczby różnych grup, dla których planowane są zajęcia, ponieważ zwiększa to czas obliczeń ocen spełnienia niektórych warunków niekrytycznych. Można zauważyć tendencję zmniejszania się rozpatrywanej prędkości przy wzroście liczby zajęć, czyli liczby poziomów drzewa przeszukiwań. Jest to spowodowane zwiększeniem średniej liczby węzłów, pomiędzy którymi trzeba przejść, aby dotrzeć do kolejnego liścia (rozwiązania).

4.2. Tabu Search

W porównaniu z algorytmem wyczerpującym, przeszukiwanie tabu jest techniką znacznie bardziej efektywną, tzn. pozwala znaleźć rozwiązania, które poprzez algorytm

wyczerpujący osiągnęlibyśmy w nieporównywalnie dłuższym czasie. Wadą tego podejścia jest to, że efekt końcowy może być gorszy niż w przypadku algorytmu dokładnego. Jednak z przeprowadzonych prób w przypadkach, w których można było zastosować także algorytm wyczerpujący wynika, że różnice w ocenie najlepszych rozwiązań znalezionych przez oba algorytmy były niewielkie. Często algorytm tabu znajdował nawet te same najlepsze rozwiązania, co algorytm wyczerpujący. Natomiast w przypadku planowania większej liczby zajęć, szukanie tabu jest jedynym z dwóch zaimplementowanych algorytmów, który może zostać użyty do uzyskania rozwiązań o zadowalającej jakości.

Lista stanów tabu spełnia swoje zadanie i pozwala opuścić osiągnięte maksimum lokalne funkcji stanu (a więc najlepsze rozwiązanie w pewnym obszarze przestrzeni przeszukiwań), by poprzez wędrówkę pomiędzy gorszymi rozwiązaniami wejść w obszar innego maksimum, dającego jeszcze większą poprawę tej funkcji, a więc jeszcze lepsze rozwiązanie. Długość wspomnianej listy powinna być na tyle duża, aby algorytm nie powracał do ostatnio odwiedzonego minimum lokalnego, ale też na tyle mała by ograniczyć liczbę nie odkrytych jeszcze stanów, do których przejście jest jednak zablokowane przez elementy listy tabu. Jak wynika z obserwacji, najlepsze wyniki zapewniała lista, w której tranzycje do poszczególnych elementów przestrzeni przeszukiwań przebywają od kilkudziesięciu do kilku tysięcy iteracji algorytmu.

Ponieważ rozwiązanie początkowe jest losowane spośród stanów przestrzeni przeszukiwań, wyniki kolejnych uruchomień szukania rozwiązań dla tego samego zadania mogą się różnić. Najczęściej różnice te są nieznaczne, jednak może się zdarzyć, że obranie innego stanu początkowego spowoduje odkrycie znacznie lepszego ekstremum funkcji stanu od tych osiągniętych po starcie z innych stanów początkowych, z których algorytm nie zdołał przejść do wspomnianego lepszego rozwiązania, mimo zastosowania listy tabu.

Czas przeszukiwania rozwiązań zależy głównie od liczby planowanych zajęć $|Z|$ oraz wielkości przestrzeni przeszukiwań $|S \times T|$. Dzieje się tak, ponieważ za każdym razem rozpatrywane jest całe sąsiedztwo aktualnego stanu, którego liczebność jest proporcjonalna do wymienionych parametrów. Czas przetwarzania nie zależy natomiast w tym przypadku od liczby "zajętych" elementów $|Z_c|$. Jest tak, ponieważ w przeciwieństwie do algorytmu wyczerpującego, algorytm tabu uwzględnia w ocenie także rozwiązania niepoprawne i może do nich przechodzić jako do kolejnych wybranych stanów. Dopuszczenie możliwości przyjmowania stanów odpowiadających harmonogramom sprzecznym pozwala na łatwiejsze przemieszanie się pomiędzy różnymi ekstremami lokalnymi funkcji stanu.

Przeszukiwanie tabu kończy się, gdy zostanie wykonana określona liczba następujących po sobie iteracji, które nie poprawiają dotychczasowego wyniku. Liczba ta jest kolejnym parametrem wpływającym znacząco na czas wykonywania procesu. Średnia wartość łącznej liczby iteracji jest ok. dwa razy większa niż wartość tego parametru. Oczywiście zwiększanie jego wartości może powodować to, że w efekcie będą znajdowane lepsze rozwiązania. Jednak dla każdych danych wejściowych, określonych głównie przez liczbę planowanych zajęć, jest pewna wartość, po przekroczeniu której, dalsze zwiększanie wartości tego parametru nie przyniesie poprawny wyniku końcowego. W przypadku planowania kilku zajęć wartość ta jest rzędu kilku tysięcy. Efekt ten jest spowodowany tym, że algorytm tabu może znaleźć najlepsze rozwiązania także dla mniejszej liczby iteracji albo w ogóle nie jest w stanie ich znaleźć z powodu zbyt krótkiej listy tabu.

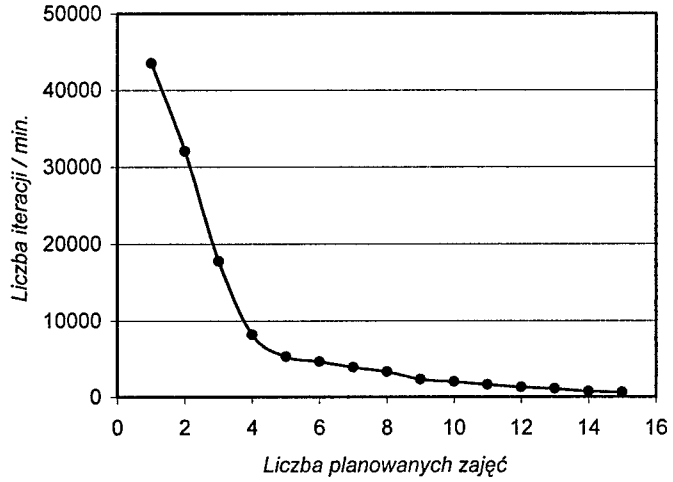
Jak wynika z powyższego, przy ustalonej maksymalnej liczbie iteracji bez postępu, czas szukania rozwiązań zależy od czasu wykonywania kolejnych iteracji. W tablicy 4.2 przedstawiona została średnia liczba wykonywanych w ciągu minuty iteracji, zależnie od liczby planowanych zajęć.

Tablica 4.2

Prędkość wykonywania iteracji tabu search w zależności od liczby planowanych zajęć.

Liczba zajęć	1	2	5	10	15	20	25
Liczba itaracji na min.	43509	32035	5324	1947	569	244	89

Ta sama zależność dla liczby zajęć nie przekraczającej piętnastu, została przedstawione na wykresie przedstawionym na rys. 4.1.



Rys. 4.1. Wykres prędkości wykonywania iteracji algorytmu przeszukiwania tabu.

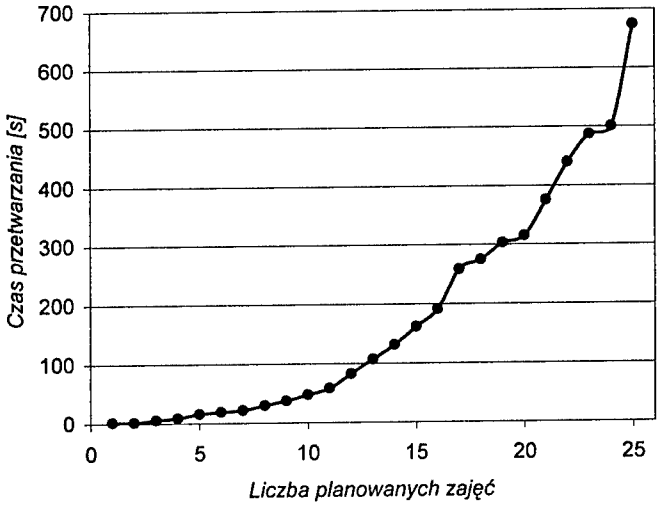
Korzystając z danych umieszczonych w tablicy 4.1 oraz z założenia, że średnia liczba wykonywanych iteracji w całym procesie jest zbliżona do podwojonej wartości maksymalnej liczby iteracji jaką może wykonać algorytm bez uzyskania poprawy, przedstawiamy szacowane czasy szukania rozwiązań w zależności od liczby planowanych zajęć oraz wartości wspomnianego parametru warunku stopu. Uzyskane wyniki zaprezentowane są w tabeli 4.3.

Tablica 4.3

Czas wykonywania algorytmu tabu search

		Maksymalna liczba iteracji bez postępu				
		500	1000	2000	5000	10000
Liczba planowanych zajęć	1	1 s.	3 s.	6 s.	14 s.	28 s.
	2	2 s.	4 s.	8 s.	19 s.	38 s.
	5	16 s.	31 s.	62 s.	156 s.	312 s.
	10	48 s.	95 s.	191 s.	476 s.	16 min.
	15	162 s.	324 s.	11 min.	27 min.	54 min.
	20	315 s.	10 min.	21 min.	52 min.	105 min.
	25	11 min.	22 min.	45 min.	112 min.	225 min.

Dane przedstawione w pierwszej kolumnie powyższej tabeli, a więc także średni czas wykonania tysiąca iteracji w zależności od liczby planowanych zajęć, przedstawiony został na rys. 4.2



Rys. 4.2. Wykres czasu wykonania tysiąca iteracji w zależności od liczby zajęć.

Z przeprowadzonej przez nas oceny harmonogramów znajdowanych przez algorytm TS wynika, że są to rozwiązania wartościowe, których osiągnięcie bez wykorzystania automatycznego planowania byłoby zadaniem pracochłonnym i trudnym. Przykładowo zaimplementowany algorytm szukania tabu jest w stanie w przeciągu niecałej minuty utworzyć plan wszystkich zajęć pewnej grupy dziekańskiej (poza przedmiotami obieralnymi), w którym nie ma żadnego okienka, zajęcia są optymalnie rozłożone w tygodniu pod kątem liczby godzin zajęć dziennie oraz odbywają się w możliwie dogodnych porach dnia.

5. EFEKTY WDROŻENIA

Architektura pakietu składa się z osobnych komponentów GUI, warstwy dostępu do bazy danych oraz algorytmów. Wykorzystywane algorytmy, zostały oprogramowane jako komponenty Delphi (VCL). Jedynym wspólnym interfejsem pomiędzy niniejszym pakietem, a pozostałymi komponentami systemu edycji i zarządzania programem studiów i planowaniem zajęć jest baza danych (wobec braku wyraźnie wydzielonej warstwy dostępu do bazy danych w istniejącym wcześniej systemie).

Pakiet ten został oddany do użytku w 2003 roku i jest obecnie z powodzeniem wykorzystywany do wspomagania układania harmonogramów zajęć na wydziale ETI PG. Na przełomie 2003/2004 przeszedł kilka kolejnych modyfikacji związanych z rozbudową interfejsu użytkownika.

Mamy nadzieję, że modularna budowa i precyzyjna dokumentacja techniczna (licząca kilkaset stron) ułatwi rozbudowę tego pakietu (np. przez wprowadzenie dodatkowych warunków niekrytycznych lub innych algorytmów heurystycznych jak np. algorytmy genetyczne) w przyszłości.

BIBLIOGRAFIA

[1] Pepol S. *Pakiet wspomagania procesu optymalizacji harmonogramu zajęć akademickich*, Praca magisterska, WETI PG, 2003.
[2] Piwakowski K., Szyfelbein D.: *Warianty algorytmu Tabu Search w zastosowaniu harmonogramów zajęć szkolnych*, Automatyka tom 7, Zesz. Nauk. Akademii Górniczo-Hutniczej w Krakowie nr 1-2, s. 229-236, 2003
[3] Piwakowski K.: *Zastosowanie metod algorytmicznych w znajdowaniu dolnych i górnych oszacowań liczb Ramseya*, Praca doktorska, WETI PG, 1996.
[4] Glover F., Taillard E., de Werra D.: *A user's guide to tabu search,* Annals of Operations Research 41, s. 3-28, 1993.
[5] Glover F., Laguna M.: *Tabu search*, Kluwer Academic Publishers, 1997.
[6] Costa D.: *A tabu search algorithm for computing an operational timetable*, European Journal of Operational Research 76, s. 98-110, 1994.
[7] Schaerf A., Schaerf M.: *Local search techniques for high school timetabling*. W: Proceedings of the 1st Intl. Conf. on the Practice and Theory of Automated Timetabling, s. 313-323, 1995.

PACKAGE FOR AIDING ACADEMIC TIMETABLES CREATION AND OPTIMIZATION

Summary

In this paper we present theoretical model and practical notes on putting package for aiding academic timetables creation and optimization into practice in ETI Department of Technical University of Gdańsk. This package contains implementation of exhaustive search and Tabu Search algorithms. In our computational model many practical quality factors are considered as for example consecutive schedules, even distribution of classes during week, dislocation of participants, etc. Beside theoretical model, we present results of practical experiments and general information on the system architecture.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Tomasz Witkowski

Katedra Algorytmów i Modelowania Systemów
Wydział Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej

PORÓWNANIE WYBRANYCH PROGRAMÓW KOMPUTEROWYCH DO UKŁADANIA ROZKŁADÓW ZAJĘĆ

Streszczenie

W artykule znajduje się porównanie dwóch komercyjnych i dwóch niekomercyjnych programów do układania rozkładów zajęć. Zostało ono dokonane w oparciu o ogólną teoretyczną wiedzę związaną z tym zagadnieniem jak i pod kątem praktycznego zastosowania. W szczególności pod uwagę wzięte zostały model danych wejściowych i narzędzia układające zajęcia. W każdym przypadku wskazane zostały zalety i wady poszczególnych rozwiązań. Na zakończenie znalazły się wyniki próby zastosowania każdego z programów do ułożenia pewnego przypadku rozkładu zajęć dla danych rzeczywistych pochodzących z Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej.

1. WSTĘP

Zagadnienie układania rozkładów zajęć jest bogato opisane w literaturze naukowej, ale też doczekało się wielu prób tworzenia aplikacji wspomagających komputerowo jego rozwiązywanie. Na rynku polskim dostępne są w szczególności propozycje firm *Vulcan* z Wrocławia i *Astar'05* z Białegostoku ([1], [2], [3], [4]). Ze względu na to, że na Wydziale Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej (dalej zwanym krótko: WETI PG) obroniony został ostatnio doktorat związany z tematyką planowania zajęć ([5]), zaś autor niniejszego artykułu jest w trakcie finalizacji pracy dyplomowej ([6]), która także tej tematyki dotyczy, zasadne wydaje się krótkie porównanie tych publikacji.

2. MODELE DANYCH WEJŚCIOWYCH

Projektowanie każdego systemu komputerowego, który ma wspomagać układanie rozkładów zajęć, należy rozpoczynać od określenia stopnia szczegółowości danych wejściowych. Odpowiedź na nie zależy od celów, jakie się przed systemem stawia. Łatwo się domyślić, że w obrębie szkół podstawowych problem istnienia odpowiednio dobrego rozwiązania jest znacznie „mniejszego kalibru" niż w przypadku szkół wyższych.

Poza sytuacjami, kiedy tworzymy system dedykowany dla konkretnej placówki, stajemy zwykle przed niełatwym zadaniem. Musimy bowiem zaproponować taki model

danych, który byłby z jednej strony jak najmniej skomplikowany, ale z drugiej strony możliwie najbardziej elastyczny i znajdował zastosowanie w jak największej liczbie przypadków.

W poniższej tabeli zgrupowane zostały informacje o modelach danych w wybranych programach komercyjnych. W obu programach operuje się na przedmiotach, nauczycielach i salach. Wyjaśnijmy jeszcze na wstępie, że przez następstwo przedmiotów rozumie się zdolność do zawarcia w modelu żądania, by pewien przedmiot odbywał się wcześniej w ciągu dnia lub tygodnia niż inny (przykładowo w planie akademickim chcemy, aby w każdym tygodniu zajęć wykład z jakiegoś przedmiotu poprzedzał ćwiczenia czy laboratoria z tegoż przedmiotu). Podział na grupy równoległe polega na tym, że w trakcie układania planu zajęć mają być utworzone grupy realizujące ten sam program nauczania z przedmiotu, na którym jest więcej osób niż mieści się w sali lub jest dopuszczalne ze względów dydaktycznych. Ten aspekt ma istotne znaczenie w przypadku planów akademickich z bardzo rozbudowaną indywidualizacją zajęć. Rozróżniać będziemy podziały ręczne (tzn. definiowane na stałe przez użytkownika) i automatyczne (tzn. tylko deklarowane przez użytkownika jako pożądane, a realizowane już przez algorytm).

Tablica 2.1

Porównanie modeli danych wejściowych w wybranych programach komercyjnych

Element modelu	*Plan lekcji* firmy *Vulcan*	*Plan lekcji* firmy *Astar'05*
Niejednostkowy czas trwania zajęć	Możliwy	Możliwy
Godziny zakazane	Możliwe dla nauczycieli i sal. Dla nauczycieli można też podawać godziny preferowane, jednakże algorytm nie musi ich przestrzegać. Nie można przyporządkowywać blokad terminów przedmiotom.	Możliwe dla nauczycieli, klas i sal. Nie można przyporządkowywać blokad terminów przedmiotom.
Następstwo przedmiotów	Można zdefiniować stopień trudności przedmiotu. Automat układający będzie się starał ułożyć plan tak, aby zajęcia trudniejsze poprzedzały łatwiejsze.	Niemożliwe
Podział na grupy równoległe	Możliwe jest w niewielkim zakresie ręczne scalanie klas lub dzielenie. Nie ma możliwości zadeklarowania przedmiotów czy grup do podzielenia automatycznego.	Duże możliwości w zakresie ręcznego dzielenia na grupy równoległe lub scalania ich, ale praktycznie brak rozwiązań automatycznych
Dodatkowe parametry	Można określić maksymalną liczbę godzin nauczyciela w ciągu dnia, maksymalna liczbę okienek nauczyciela w dniu i tygodniu. Można też dla pewnych nauczycieli określić te parametry oddzielnie, uwzględniając np., że któryś z nich nie może mieć w ogóle okienek.	Nie ma

Teraz w podobny sposób dokonamy porównania wybranych prac o charakterze niekomercyjnym ([5], [6]). Wyjaśnijmy jednak od razu, że pierwsza z nich miała charakter

raczej dedykowany (była dostosowana do realiów WETI PG), natomiast druga ma z założenia traktować zagadnienie układania rozkładów bardziej ogólnie i elastycznie.

Tablica 2.2

Porównanie modeli danych wejściowych w wybranych programach niekomercyjnych

Element modelu	Praca doktorska D. Szyfelbeina	Praca magisterska T. Witkowskiego (w przygotowaniu)
Niejednostkowy czas trwania zajęć	Możliwy	Możliwy
Godziny zakazane	Możliwe jest zakazanie godziny dla nauczycieli i grup studenckich oraz nakazanie terminu dla zajęć (wstępne przyporządkowanie).	Możliwe dla każdej osoby.
Następstwo przedmiotów	Niemożliwe	Niemożliwe
Podział na grupy równoległe	Tylko ręczny	Ręczny lub automatyczny
Dodatkowe parametry	Nie ma	Możliwość przydzielenia do 3 sal jednemu przedmiotowi (tak, aby mógł się on odbywać w więcej niż jednej sali równocześnie)

3. ALGORYTMY UKŁADAJĄCE PLANY ZAJĘĆ

3.1. Zarys metod służących do rozwiązywania problemu układania rozkładów zajęć

Dysponując modelem danych wejściowych, musimy zdefiniować, co dokładnie będziemy rozumieli pod pojęciem planu zajęć i jakimi metodami będziemy go poszukiwać. Wybór ten ma kluczowy wpływ na jakość całego algorytmu. Zwykle mamy w tym zakresie do dyspozycji ten sam zestaw narzędzi teoretycznych, w skład których wchodzą między innymi ([5], [6], [7], [8], [9], [10]):

- rozmaite strategie zachłanne (w szczególności wywodzące się z różnych wariantów kolorowania grafów lub szeregowania zadań);
- rozmaite algorytmy przeszukiwania wyczerpującego i pochodne od nich (np. przeszukiwanie z nawrotami);
- algorytmy programowania liniowego (najczęściej całkowitoliczbowego);
- heurystyki zaczerpnięte ze sztucznej inteligencji, w szczególności algorytmy genetyczne, metody symulowanego wyżarzania lub przeszukiwania z listą tabu.

3.2. Wymagania stawiane algorytmom i kryteria ich oceny

Każdy plan, co oczywiste, musi respektować tzw. „twarde" ograniczenia (ang. *hard constraints*), takie jak brak konfliktów czasowych między zajęciami tej samej osoby, umieszczenie co najwyżej jednych zajęć w danej sali na danej godzinie itd.

W szczególności bardzo istotne jest tutaj podanie kryteriów optymalności planu. Kryteria te mogą być rozmaite. W polskich realiach najczęściej rozpatruje się tutaj ([1], [2], [3], [4], [5], [6]):

- ograniczoną liczbę okienek;
- równomierność rozłożenia zajęć w ciągu tygodnia;
- unikanie umieszczania zajęć na skrajnych godzinach (wczesnych rannych lub późnych popołudniowych);
- unikanie umieszczanie bloków tego samego przedmiotu więcej niż raz w ciągu dnia.

Łatwo można zauważyć, że wystąpienie pewnej korzystnej z powyższego punktu widzenia cechy planu zajęć może pociągać za sobą występowanie również innych korzystnych cech i na odwrót, np. w planie o niewielkiej liczbie okienek najczęściej wystąpi tendencja do niewielkiego zapełnienia szkoły na skrajnych rannych lub popołudniowych godzinach w ciągu dnia i zbliżonego do maksymalnego – na godzinach południowych.

Trzeba jednak powiedzieć, że w niektórych publikacjach zachodnich ([7]) z uwagi na spore różnice w systemie szkolnictwa można spotkać zupełnie inne kryteria, np.:

- unikanie przydzielania jednych zajęć na cały dzień;
- unikanie przydzielania więcej niż dwóch zajęć pod rząd bez okienka (!).

Dodatkowo pożądanymi cechami algorytmów układających plany zajęć są:

- sprawdzenie niesprzeczności danych wejściowych (tzn. pewnych warunków koniecznych do istnienia rozwiązania) tak, aby program potrafił wykrywać sytuacje, kiedy nie da się znaleźć żadnego rozwiązania dopuszczalnego;
- układanie 100% przedmiotów w jednej lub kilku turach w taki sposób, aby nie była wymagana ingerencja operatora (np. ręczne poprawianie w końcówce pracy);
- budowa umożliwiająca wykorzystanie wiedzy osoby doświadczonej w układaniu rozkładów w konkretnej szkole (w szczególności może to być związane z szeroką gamą narzędzi wspomagających nadzór nad kolejnością umieszczania zajęć i tworzeniem planów częściowych);
- szybkość działania.

Przy okazji naturalnie pojawia się pytanie o sposoby porównywania planów zajęć ułożonych za pomocą różnych programów. W przypadku, gdy programy te dokonują optymalizacji pod kątem istotnie różnych kryteriów, takie porównanie jest bardzo utrudnione. Jeżeli jednak owe kryteria są podobne lub najlepiej takie same, możliwe są różne sposoby postępowania, np.:

- uruchomienie obu programów na tych samych danych i na tej samej maszynie i dokonanie bezpośredniej konfrontacji;
- napisanie programu wzorcowego (ang. *benchmark*), którego tempo pracy i jakość wyników stanowią pewien poziom odniesienia i następnie wywołanie tego programu i programu testowanego na tych samych danych i tej samej maszynie w celu otrzymania informacji o względnej przewadze programu testowanego ([7]).

3.3. Najważniejsze zalety i wady działania wybranych programów komercyjnych

Mając w pamięci powyższe uwagi, skomentujemy działanie algorytmów w omawianych programach komercyjnych.

Oba programy potrafią zbadać podstawowe warunki wymagane do niesprzeczności danych, żaden z nich nie układa planu zajęć do końca, tzn. zawsze jest wymagana ingerencja operatora. Pomimo iż dysponujemy pewną liczbą narzędzi wspomagających te

końcowe czynności (np. mechanizm wyszukiwania możliwości zamian w planie), tego rodzaju zachowanie należy traktować jako wysoce niepożądane, szczególnie w przypadku dużych placówek. W programach można zadeklarować chęć tworzenia planu częściowego, pomijającego niektóre przedmioty i ewentualnie uzupełniającego wcześniejszy plan częściowy. Poza tym można zaobserwować tendencję polegającą na tym, że programy, dysponując pełnym zestawem przedmiotów, pozostawiają na koniec przedmioty najtrudniejsze do ułożenia. Innymi słowy, chcąc z nich sensownie korzystać, należy układać rozkład zajęć w kilku turach, za każdym razem selekcjonując samodzielnie przedmioty do ułożenia w oparciu o wiedzę o realiach organizacyjnych szkoły.

Algorytmy komercyjne są szybkie – można założyć, że pracując z małymi porcjami danych, plan lub jego część powstaje w zasadzie momentalnie. Jeżeli jednak weźmiemy pod uwagę czas, jaki trzeba poświęcić na analizę, ręczne poprawki i uzupełnianie otrzymywanych rozwiązań, może się okazać, że ta szybkość jest tylko pozorna.

W przypadku programu firmy *Vulcan* za najważniejsze wady można uznać niemożność automatycznego rozłożenia zajęć podzielonych na więcej niż 2 grupy oraz scalania lekcji dzielonych, ale mających różny czas trwania. Poza tym w przypadku wieloetapowego tworzenia planu w kolejnych etapach nie można przesuwać bloków wielogodzinnych ułożonych w etapach wcześniejszych, a zatem zachodzi domniemanie, że przy dużej liczbie takich bloków jakość rozwiązania znacznie spada. Ponadto zajęcia łączące grupy (części klas) są rozmieszczane zawsze na pierwszej lub ostatniej lekcji w dniu.

Program firmy *Astar'05* jest zbyt podatny na kolejność podawania przedmiotów do ułożenia i ma tendencję do tworzenia sporej liczby okienek w przypadku, gdy w jednej turze tych przedmiotów do ułożenia jest zbyt wiele.

Należy jednak sprawiedliwie oddać, że wymienione sytuacje, w których komercyjne algorytmy zawodzą, są rzadko spotykane w praktyce i objawiają się jedynie w skomplikowanych testach, o czym będzie jeszcze mowa w dalszej części tego tekstu.

3.4. Najważniejsze zalety i wady działania wybranych programów niekomercyjnych

W obu omawianych pozycjach przywiązuje się szczególną wagę do rozwiązywania problemu planowania zajęć w przypadku szkół wyższych, dzięki czemu programy te lepiej sobie radzą z rozmaitymi anomaliami, jakie są charakterystyczne dla tego typu zagadnień. Oczywiście nie wyklucza to w żadnym stopniu użycia zaproponowanych w nich metod do prostszych przypadków. Wśród wad można wymienić mniejsze niż w przypadku pakietów komercyjnych przywiązanie do punktu widzenia operatora systemu – nierzadko do praktycznego użytkowania konieczna jest pewna wiedza wykraczająca poza sprawy planistyczne (np. wiedza o naturze działania zastosowanego algorytmu).

3.5. Porównanie działania programów komercyjnych i niekomercyjnych

W porównaniu z programami firm *Vulcan* i *Astar* obie wybrane pozycje niekomercyjne mają przede wszystkim tę przewagę, że radzą sobie znacznie lepiej z bardziej skomplikowanymi przypadkami planów akademickich. Jest to oczywiście ze wszech miar zrozumiałe, chociażby z tego względu, że zdecydowaną większość stanowią placówki, w których ułożenie planu zajęć nie jest zadaniem aż tak skomplikowanym jak np. na WETI PG. Z drugiej strony komercja wymusza stosowanie wielu ciekawych i praktycznych rozwiązań w zakresie możliwości prezentacji rozwiązania (np. generuje się automatycznie „płachty" czy strony HTML z planem zajęć) oraz powoduje, że program układający plan zajęć jest rozsądnie zintegrowany z innymi programami wspomagającymi zarządzanie

szkołą. Szczególnie ta druga cecha ma bardzo istotne znaczenie z praktycznego punktu widzenia.

4. WYDAJNOŚĆ PROGRAMÓW W PRZYPADKU UKŁADANIA ROZKŁADU ZAJĘĆ DLA WETI PG

Omawiane w tej części tekstu wyniki dotyczą testu polegającego na ułożeniu rozkładu zajęć jedynie dla kierunku informatyka. Dane do testu były zgodne z rzeczywistymi dla semestru zimowego roku akademickiego 2001/02. Wyjątek stanowi praca [5] – odniesiono się tutaj do wyników zamieszczonych w niej przez Autora dla testów o zbliżonym stopniu trudności.

Okazuje się, że programem firmy *Vulcan* nie da się rozwiązać tak postawionego problemu. Wynika to z braku możliwości definiowania podziałów na grupy równoległe, ścisłe przywiązanie do pojęcia klasy i nauczyciela, a także na przyjęte dość ciasne ograniczenia rozmiaru danych wejściowych. Nie oznacza to bynajmniej, że program ten jest jednoznacznie najsłabszym z omawianych, gdyż w prostszych sytuacjach daje on z kolei spore możliwości, np. jako jedyny ma mechanizm umożliwiający wpływ użytkownika na kolejność rozłożenia zajęć w ciągu dnia.

Lepiej w teście wypada program firmy *Astar'05*. Za jego pomocą jest możliwe ułożenie rozkładu zajęć we wspomnianym przypadku. Trudno jednak wydać racjonalną ocenę jakości tego rozwiązania, gdyż do jego wygenerowania potrzebne było wielokrotne wywoływanie automatu dla małych porcji danych, a zatem intuicja i wiedza planistyczna operatora systemu ma tutaj bardzo istotne znaczenie.

Oba programy niekomercyjne są w stanie ułożyć plan zajęć dla WETI PG w czasie nie większym niż kilka minut (czas potrzebny na wprowadzanie danych oczywiście pomijamy, chociaż z praktycznego punktu widzenia ma on ogromne znaczenie). Czas ten jest w gruncie rzeczy zależny od wielu czynników: w przypadku [5] można samemu określić jedną z 3 metod układania rozkładu zajęć (algorytm genetyczny, symulowane wyżarzanie lub przeszukiwanie tabu), zaś w przypadku [6] dopuszczalne jest przerwanie pracy w każdym momencie (np. gdy zauważymy, że przez pewien czas nie ma postępu w jakości znajdowanych rozwiązań). Jakość rozwiązań jest zbliżona, przy czym program [5] ma tę przewagę, że w trakcie znajdowania rozwiązań bierze pod uwagę równomierność rozłożenia zajęć w ciągu tygodnia.

5. ZAKOŃCZENIE

Wydaje się, że pomimo bardzo szybkiego rozwoju technologii informatycznych nadal istnieje spore zapotrzebowanie na oprogramowanie wspomagające układanie rozkładów zajęć. W wielu szkołach bowiem nie korzysta się z żadnych narzędzi komputerowych ułatwiających ten proces. Z drugiej strony obecnie pojawiła się tendencja do tworzenia dużych systemów informatycznych (np. pakiet *Uczelnia XP*), które łączą w sobie administracyjne usługi leżące zwykle w gestii dziekanatów z planistycznymi i mnóstwem innych. Jak pokazują powyższe rozważania, trudnym zadaniem jest stworzenie w pełni funkcjonalnego systemu do układania rozkładów zajęć, a w szczególności takiego, który byłby w stanie sprostać wymaganiom organizacyjnym uczelni wyższych.

BIBLIOGRAFIA

[1] Strona internetowa firmy *Vulcan*: www.vulcan.pl
[2] Podręcznik użytkownika programu *Plan lekcji* firmy *Vulcan*, Wrocław 1994
[3] Strona internetowa firmy *Astar'05*: www.astar05.bial.pl
[4] Podręcznik użytkownika programu *Plan lekcji* firmy *Astar'05*, Białystok 2002
[5] Szyfelbein D.: *Metody lokalnego i globalnego przeszukiwania przestrzeni rozwiązań dla problemu układaniu rozkładów zajęć*, rozprawa doktorska WETI PG 2003
[6] Witkowski, T.: *Algorytm układania rozkładów zajęć dla szkół wyższych*, praca magisterska (na etapie finalizacji)
[7] Chiarandini M., Socha K., Birattari M., Rossi Doria O.: *An effective hybrid approach for the university course timetabling problem*, Metaheuristics Network 2003
[8] Burke E., Ross P.: *Practice and theory of automated timetabling, First International Conference, Edinburgh*, Springer 1996
[9] Burke E., Carter M.: *Practice and theory of automated timetabling II, Second International Conference, Toronto*, Springer 1998
[10] Burke E., Erben W.: *Practice and theory of automated timetabling III, Third International Conference, Konstanz*, Springer 2000.

COMPARISON OF SELECTED COMPUTER PROGRAMS FOR AUTOMATED TIMETABLING

Summary

In the article two commercial and two noncommercial programs for computer-aided timetabling are compared. It is done according to both theoretical knowledge about such a problem and practical use. In particular the input data model and efficiency of the timetabling tools were taken into account. In each case advantages and disadvantages were shown. Finally, the results of the test, based on real data from the Electronics, Telecommunications and Informatics Faculty of Gdansk University of Technology, are presented.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Bogdan Bartosiński

Katedra Metrologii i Systemów Elektronicznych, Politechnika Gdańska

STANOWISKO LABORATORYJNE DO TESTOWANIA ANALOGOWYCH UKŁADÓW ELEKTRONICZNYCH Z WYKORZYSTANIEM MAGISTRALI IEEE 1149.4

Streszczenie

Przedstawiono stanowisko laboratoryjne do testowania analogowych układów elektronicznych z wykorzystaniem magistrali testującej mieszanej sygnałowo zgodnej ze standardem IEEE 1149.4. Stanowisko laboratoryjne zorganizowano w oparciu o pierwsze komercyjne układy wyposażone w magistralę IEEE 1149.4 – układy scalone STA400, opracowane przez firmę National Semiconductor i Logic Vision. Sterowanie magistralą odbywa się poprzez kontroler wykonany z wykorzystaniem portu równoległego komputera PC. Program sterujący kontrolerem został zrealizowany w środowisku LabWindows/CVI 5.0. Stanowisko umożliwia testowanie układów analogowych metodami opisanymi w normie IEEE 1149.4, a także badania nowych metod diagnostyki ukierunkowanych na testowanie magistralowe.

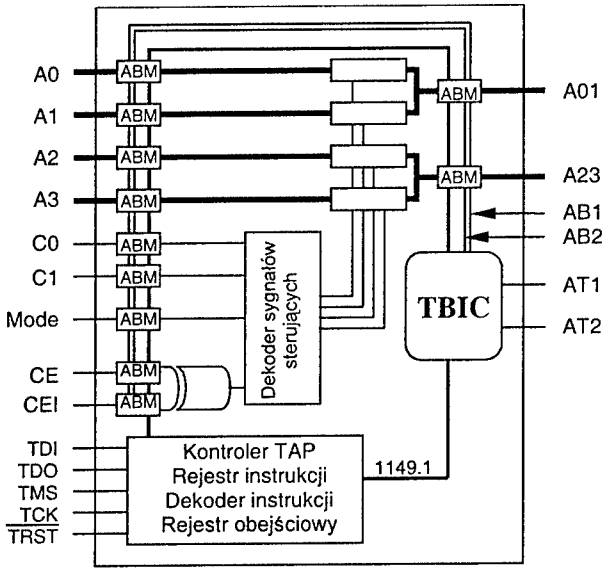
1. WSTĘP

Standard magistrali testującej mieszanej sygnałowo IEEE 1149.4 [1] został opracowany w 1999 r. Jest on rozszerzeniem standardu magistrali ułatwionego testowania dla układów cyfrowych IEEE 1149.1 [2]. Podstawowym zadaniem standardu jest umożliwienie dostępu pomiarowego poprzez magistralę analogową do układów scalonych na pakiecie, co pozwala na uproszczenie lub wyeliminowanie z procesu testowania kosztownych głowic ostrzowych. Standard zapewnia następujące rodzaje testowania:

- testowanie połączeń – poszukiwanie zwarć i rozwarć wśród połączeń na pakiecie,
- testowanie parametryczne – pomiar charakterystyk układów analogowych i testowanie obecności i wartości elementów dyskretnych,
- testowanie wewnętrzne – testowanie rdzenia elementu mieszanego sygnałowo.

2. IMPLEMENTACJA MAGISTRALI IEEE 1149.4 W UKŁADACH STA400

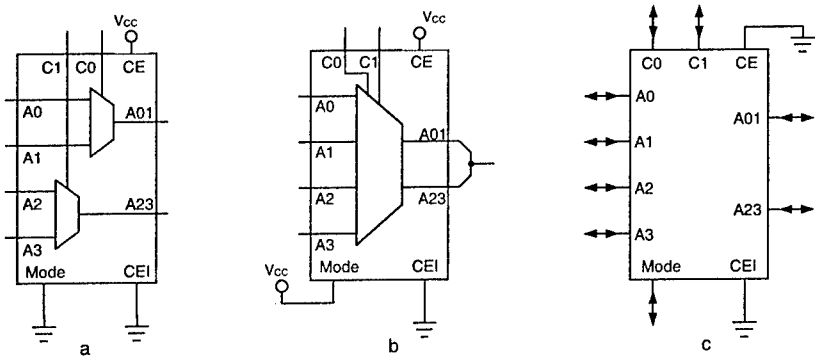
Układ STA400 [3] firmy National Semiconductor i Logic Vision jest pierwszym komercyjnym układem w pełni wyposażonym w mieszaną sygnałowo magistralę testującą IEEE 1149.4. Schemat blokowy układu przedstawiony jest na rys. 1.



Rys. 1. Schemat blokowy układu STA400.

Rdzeń układu, czyli część realizująca jego funkcje użytkowe, składa się czterech kluczy analogowych o małej rezystancji oraz układu dekodera odpowiadającego za włączenie odpowiedniego klucza. Układ można skonfigurować jako dwa dwuwejściowe analogowe multipleksery lub jeden multiplekser czterowejściowy.

Pomiędzy wyprowadzeniami funkcjonalnymi układu a rdzeniem znajdują się analogowe moduły brzegowy ABM. Analogowe sygnały magistrali doprowadzane są liniami AT1, AT2 i poprzez interfejs szyny testowej TBIC łączą się z wewnętrzną magistralą analogową AB1, AB2, która dochodzi do poszczególnych modułów ABM.



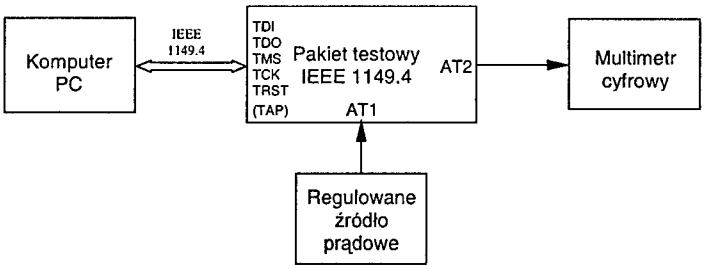
Rys. 2. Przykłady konfiguracji układu STA 400: a) podwójny dwuwejściowy multiplekser, b) pojedynczy czterowejściowy multiplekser, c) wirtualna sonda pomiarowa.

Dziewięć modułów ABM może być wykorzystanych w charakterze wirtualnych sond pomiarowych. Sondy te pozwalają obserwować poprzez magistralę na wyprowadzeniach układu analogowe sygnały, lub wymuszać analogowe pobudzenia. Przykłady konfiguracji układu STA 400 pokazane są na rys. 2.

Układ STA400 jest wyposażony w trzy podstawowe rejestry: instrukcji, obejściowy i brzegowy. Układ obsługuje wszystkie instrukcje obowiązkowe standardu IEEE 1149.4 oraz dodatkowo kilka instrukcji opcjonalnych.

3. OPIS STANOWISKA LABORATORYJNEGO

Schemat blokowy stanowiska pomiarowego jest przedstawiony na rys. 3.

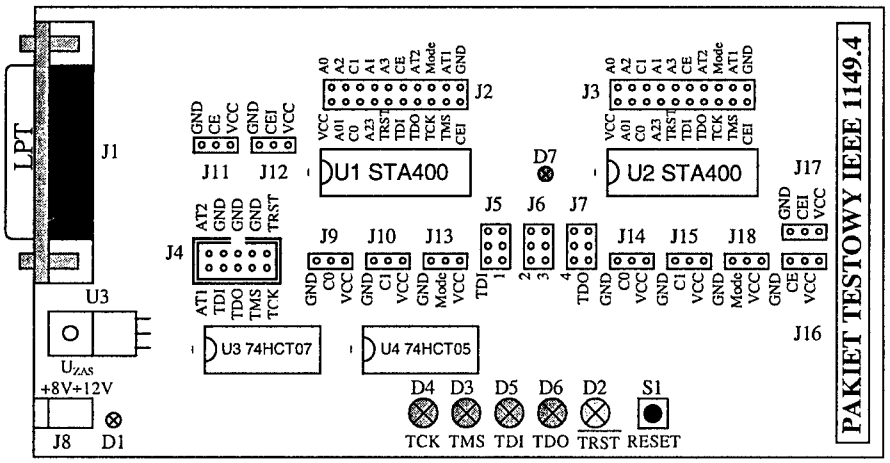


Rys. 3. Schemat blokowy stanowiska pomiarowego.

W skład stanowiska wchodzi kontroler magistrali IEEE 1149.4 zrealizowany z wykorzystaniem portu równoległego komputera PC, pakiet testowy, źródło wymuszeń prądowych oraz multimetr cyfrowy.

3.1. Pakiet testowy

Pakiet testowy pokazany jest na rys. 4. Jest on zrealizowany w oparciu o dwa układy U1 i U2 STA400 wyposażone w magistralę IEEE 1149.4.



Rys. 4. Pakiet testowy.

Wszystkie wyprowadzenia układów STA400 są dostępne pomiarowo, poprzez dołączenie ich do dwóch złącz J2 i J3. W złączach można bezpośrednio umieszczać płytki

z badanymi układami analogowymi zarówno konwencjonalnymi, jak też wyposażonymi w magistralę. System zworek J5 – J7 służy do konfiguracji łańcucha ścieżki brzegowej. Do ustalania konfiguracji i trybu pracy układów STA400 służy pole zworek J9 – J18.

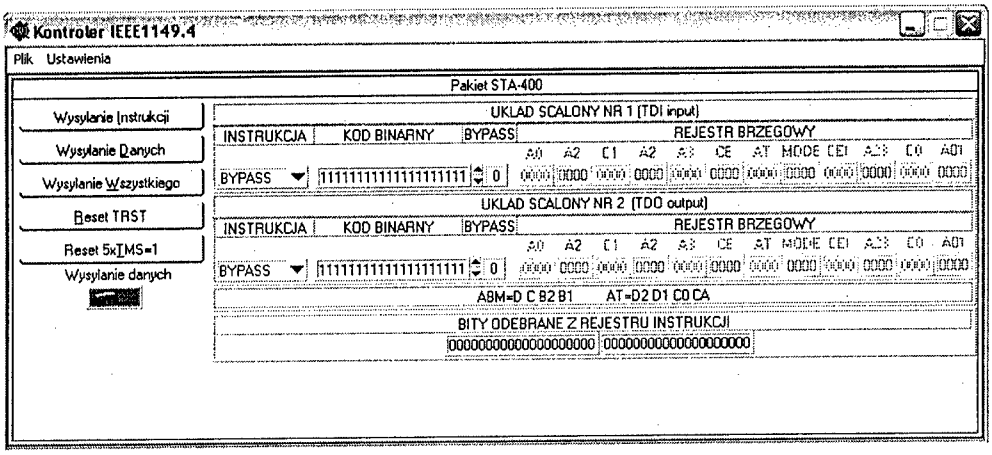
Sterowanie portu TAP odbywa się za pośrednictwem kontrolera zrealizowanego z wykorzystaniem portu równoległego komputera PC. Port TAP posiada cztery linie TDI, TDO, TMS, TCK wymagane przez standard IEEE 1149.4 oraz jedną linię opcjonalną $\overline{\text{TRST}}$. Stan linii portu TAP jest monitorowany za pomocą diod LED, dołączonych do magistrali przez inwertery z otwartym drenem typu 74HCT05. Ze względu na różne poziomy napięć w portach równoległych komputerów, pomiędzy liniami portu równoległego a liniami portu TAP zastosowano bufor z otwartym drenem typu 74HCT07.

Z powodu dużej rezystancji kluczy magistrali, stosowane metody diagnostyczne wymagają używania przede wszystkim wymuszeń prądowych. W stanowisku wykorzystane jest źródło prądowe DC o regulowanej wartości prądu w zakresie 100 – 1000 μA. W celu zabezpieczenia przed przekroczeniem dopuszczalnych napięć na wejściach układu STA400 źródło prądowe posiada ograniczenie napięciowe.

3.2. Kontroler magistrali IEEE 1149.4

Kontroler magistrali został zrealizowany programowo z wykorzystaniem portu równoległego komputera. Do sterowania liniami portu TAP badanych układów kontroler wykorzystuje cztery linie wyjściowe portu równoległego i jedną linię wejściową. Program kontrolera został napisany w środowisku LabWindows/CVI 5.0.

Kontroler realizuje podstawowe funkcje niezbędne do sterowania magistralą i do analizy odpowiedzi na testy cyfrowe. W szczególności kontroler ma możliwość wysyłania kodów instrukcji do rejestru instrukcji układów STA400 i odczytu słowa statusu z tego rejestru oraz wysyłania danych konfiguracyjnych do rejestru brzegowego i odczytu z tego rejestru. Wygląd panelu głównego programu kontrolera pokazany jest na rys. 5.



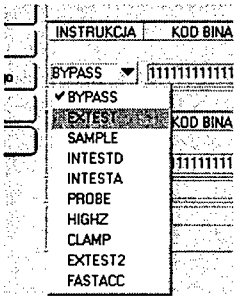
Rys. 5. Panel główny programu kontrolera magistrali.

Z lewej strony panelu głównego znajdują się przyciski służące do wysyłania sygnału zerowania oraz przyciski wysyłające instrukcje i dane do odpowiednich rejestrów.

Przycisk ***Wysyłanie Instrukcji*** pozwala na wysłanie do układów STA400 kodu aktualnej instrukcji i jednoczesne odczytanie statusu z rejestru instrukcji.

Wciśnięcie przycisku ***Wysyłanie Danych*** powoduje wysłanie do rejestru brzegowego bądź do rejestru obejściowego wcześniej ustawionych w odpowiednich polach panelu głównego danych. Jednocześnie jest odczytywana zawartość rejestru brzegowego obrazująca m.in. stan komparatorów w modułach ABM i układzie TBIC.

Wybór instrukcji wysyłanej do układu odbywa się przy pomocy menu przedstawionego na rys. 6. Obok nazwy wybranej instrukcji wyświetlany jest jej kod binarny. Instrukcją domyślną ustawianą po włączeniu programu jest instrukcja BYPASS.

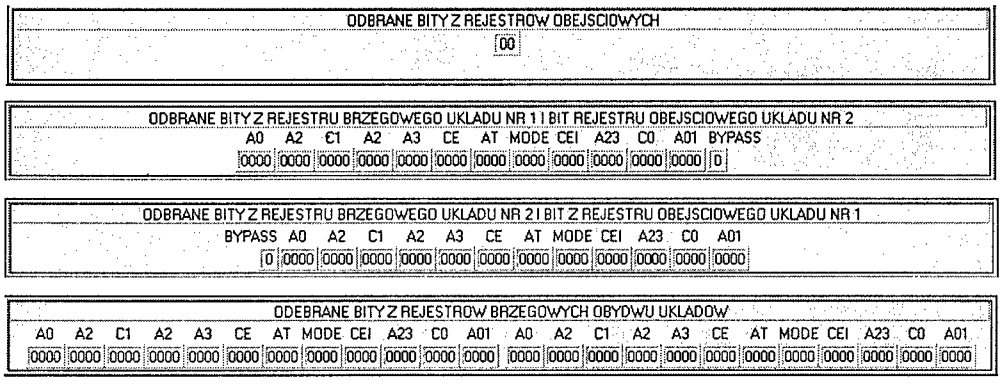


Rys. 6. Wybór instrukcji.

Prawa część panelu głównego zawiera pola, w których wpisuje się wartości bitów, które mają zostać wysłane do rejestru brzegowego układu. Wartości bitów w tych polach określają konfigurację modułów ABM i układu TBIC. W panelu umieszczone są dwa komplety takich pól, co pozwala niezależnie konfigurować obydwa układy na pakiecie.

Poniżej pól do wprowadzania danych do rejestrów brzegowych znajdują się dwa pola, w których jest wyświetlany dwudziestobitowy status odczytywany z rejestru instrukcji. podczas wysyłania kodu nowej instrukcji.

Na samym dole panelu głównego znajduje się panel pokazujący zawartość rejestrów brzegowych lub obejściowych odczytaną w trakcie wysyłania do nich danych. Panel ten jest przedstawiony na rys. 7 dla czterech możliwych konfiguracji ścieżki brzegowej.



Rys. 7. Widok panelu z zawartością rejestrów brzegowych i rejestrów obejściowych.

Informacja z tego panelu może być wykorzystywana m.in. w trakcie testowania połączeń, gdzie wybrane bity rejestru brzegowego obrazują stan komparatorów w interfejsie TBIC i modułach ABM.

Poniżej przycisków znajduje się komponent typu LED, który ma kolor zielony, gdy nie są wysyłane dane, a czerwony gdy dane są wysyłane do pakietu.

Istnieje możliwość rejestracji dokonywanych operacji przesłań instrukcji i danych do/z pakietu testowego. Opcję tą włącza się w menu ***Plik***, przez zaznaczenie pola ***Tryb rejestracji***. Wysyłane instrukcje, odczytany status, wysyłane dane i bity odczytane z rejestrów danych są zapisywane w pliku domyślnym umieszczonym w katalogu, w którym znajduje się program lub też w pliku wskazanym przez użytkownika w podmenu ***Wybierz plik***.

Poniżej pokazana jest zawartość przykładowego pliku z zarejestrowanymi operacjami wykonywanymi przez użytkownika:

```
KONTROLER IEEE 1149.4
RESET
I-OUT: 00000000000000000000111111111111111111111
I-IN: 00000000000000010101000000000000000010101
D-OUT: 00000100000100000000000011000000000000000000000001
D-IN: 10100101011100010100010011010000110000000001000100
TRST
```

W pierwszej linii pliku zamieszczany jest nagłówek, taki sam we wszystkich plikach. Kolejna linia zawiera słowo ***RESET*** oznaczające, że dokonano resetu układu za pomocą przycisku ***Reset 5xTMS=1***. Słowo ***TRST*** oznacza również reset układu, ale przy pomocy linii $\overline{\text{TRST}}$. Literka ***I*** na początku linii oznacza, że bity podane po dwukropku dotyczą rejestru instrukcji, a literka ***D*** oznacza któryś z rejestrów danych. Jeśli po myślniku występuje słowo ***OUT*** oznacza to, że wartości bitów podane po dwukropku były wysłane do pakietu z programu, a słowo ***IN*** oznacza, iż są to dane odczytane z pakietu. W jednej linii umieszczane są zawartości rejestrów dwóch układów wchodzących w skład pakietu. Najpierw jest podawana zawartość rejestrów układu nr 1 a później układu nr 2.

W podanym powyżej przykładzie w linii I-OUT do układu nr 1 wysyłana jest instrukcja EXTEST (kod 00000000000000000000), do układu nr 2 instrukcja BYPASS (kod 11111111111111111111) a w linii I-IN odczytywany jest status.

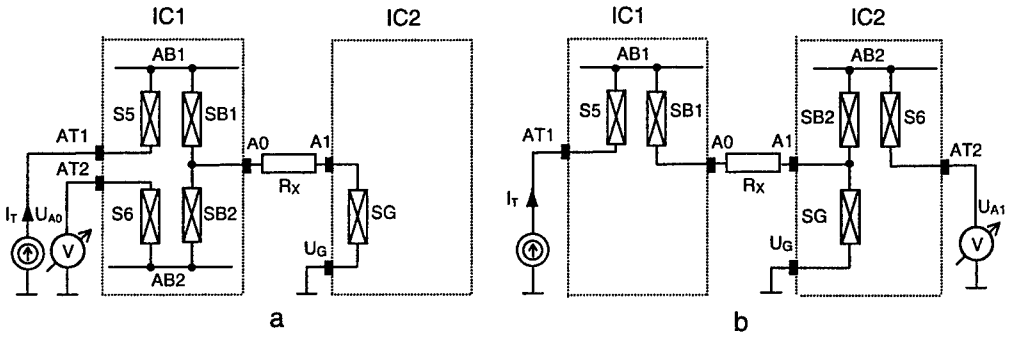
Następnie w linii D-OUT do rejestru brzegowego układu nr 1 wysyłane jest 48 bitów danych "000001000001000000000000011000000000000000000000000" a do rejestru obejściowego układu nr 2 jednobitowa dana "1". W linii D-IN odczytywana jest poprzednia zawartość rejestru brzegowego, obrazująca stan układu przed wysłaniem danych.

4. PRZYKŁADY POMIARÓW WYKONYWANYCH PRZY POMOCY STANOWISKA LABORATORYJNEGO

Zrealizowane stanowisko znalazło zastosowanie w laboratorium studenckim jak również jest wykorzystywane do praktycznej weryfikacji metod diagnostyki ukierunkowanych na testowanie układów elektronicznych z wykorzystaniem magistrali mieszanej sygnałowo IEEE 1149.4. Poniżej, na rys. 8, zaprezentowano przykład wykorzystania magistrali do pomiaru rezystancji elementu umieszczonego pomiędzy wyprowadzeniami układów STA400 z użyciem wykonanego stanowiska. Bardziej złożone metody testowania i diagnostyki z wykorzystaniem magistrali zostały opisane w pracach [4-7].

Proces pomiarowy składa się z dwóch etapów. W pierwszym etapie (rys. 8a) pobudzenie ze źródła prądowego przez linię AT1 i klucz S5 interfejsu TBIC i klucz SB1 modułu ABM dochodzi do węzła A0 pierwszego układu, połączonego z rezystorem R_X, a następnie poprzez węzeł A1 układu IC2, klucz SG modułu ABM i węzeł V_G do masy. Napięcie w węźle A1 mierzone jest woltomierzem połączonym z linią AT2, po przejściu przez klucze SB2 i S6.

W etapie drugim (rys. 8b) pobudzenie prądowe jest dołączone analogicznie jak w poprzednim etapie, natomiast napięcie jest mierzone w węźle A1 drugiego układu za pośrednictwem kluczy SB2 i S6 i linii AT2.



Rys. 8. Pomiar rezystancji elementu pomiędzy wyprowadzeniami układów scalonych z wykorzystaniem magistrali IEEE 1149.4

Jeżeli

$$R_V + R_{S6} + R_{SB2} \rangle\rangle R_{SG} \tag{4.1}$$

gdzie: R_{S6}, R_{SB2}, R_{SG}, R_V odpowiednio rezystancje kluczy magistrali i woltomierza

to cały prąd będzie przepływał do węzła U_G i gdy

$$R_V \rangle\rangle R_{S6} + R_{SB2} \tag{4.2}$$

to U_{A0}, U_{A1} będą dobrym przybliżeniem napięcia w węźle A0 i A1. W takim przypadku mierzona rezystancja R_X jest dana wyrażeniem

$$R_X = \frac{U_{A0} - U_{A1}}{I_T} . \tag{4.3}$$

5. PODSUMOWANIE

Zrealizowane stanowisko laboratoryjne do testowania analogowych układów elektronicznych testowania pozwoliło na przeprowadzenie badań właściwości metrologicznych magistrali mieszanej sygnałowo oraz na zbadanie metod pomiarowych ukierunkowanych na testowanie z wykorzystaniem magistrali. Wykonane badania potwierdziły użyteczność układów STA400 wyposażonych w magistralę IEEE 1149.4 do diagnostyki układów

analogowych. Układy te z powodzeniem mogą być zastosowane do pomiarów z dokładnością od ułamka procenta do pojedynczych procentów kilkuelementowych struktur rezystancyjnych. Badania wykazały również wady układów STA400, a przede wszystkim dużą rezystancję kluczy magistrali analogowej oraz błędy pomiaru napięć (na poziomie kilku miliwoltów) w module ABM, który jest dołączony kluczem SG do masy. W dalszym etapie badania powinny objąć pomiary zmiennoprądowe elementów RLC z wykorzystaniem magistrali.

BIBLIOGRAFIA

[1] *IEEE Standard for a Mixed-Signal Test Bus.* IEEE Std 1149.4-1999.
[2] *IEEE Standard Test Access Port and Boundary- Scan Architecture.* IEEE Std 1149.1-1990.
[3] Filliter K.., Duzevik I.: *Mixed Signal Testing Using the IEEE 1149.4 STA400.* National Semiconductor Application Note 1200. Dezember 2001.
[4] Toczek W., Bartosiński B.: *Strategie testowania układów analogowych z wykorzystaniem magistrali IEEE 1149.4.* Materiały konferencyjne V Szkoła-Konferencja Metrologia Wspomagana Komputerowo, MWK'2001, Rynia k/Warszawy, 21-24 maja 2001, t.3, s.165-170.
[5] Bartosiński B.: *Diagnostyka układów analogowych z wykorzystaniem magistrali testującej mieszanej sygnałowo IEEE 1149.4.* Pomiary Automatyka Kontrola 2002, nr 7/8, s. 17-20.
[6] Bartosiński B.: Zastosowanie układów testowych STA 400 z magistralą testującą mieszaną sygnałowo IEE 1149.4 do diagnostyki układów analogowych. Materiały konferencyjne VI Szkoła-Konferencja Metrologia Wspomagana Komputerowo, MWK'2003, Waplewo, 26-29 maja 2003, t. 2, s.209-214.
[7] Zaranek W.: *Diagnostyka układów analogowych z wykorzystaniem magistrali IEEE 1149.4.* Praca dyplomowa, Wydział Elektroniki Telekomunikacji i Informatyki Politechniki Gdańskiej, Gdańsk, 2002.

LABORATORY STAND FOR TESTING ANALOG ELECTRONIC CIRCUITS USING MIXED SIGNAL TEST BUS IEEE 1149.4

Summary

The laboratory stand for testing analog circuits using mixed signal test bus IEEE 1149.4 is presented. The stand is organized on the base of first general purpose IEEE 1149.4 compliant integrated circuits STA400, produced by National Semiconductor and Logic Vision. The main part of stand, IEEE 1149.4 controller, is made using parallel port of computer PC. The controller software has been made in LabWidows/CVI 5.0 environment. With this laboratory stand it is possible to made diagnosis of analog circuits by methods described in IEEE 1149.4 Standard and investigate new diagnostic methods with analog test bus.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Jacek Jakusz, Waldemar Jendernalik

Katedra Systemów Mikroelektronicznych, Politechnika Gdańska

ANALOGOWE FILTRY PASMA PODSTAWOWEGO SCALONYCH ODBIORNIKÓW TELEFONII KOMÓRKOWEJ

Streszczenie

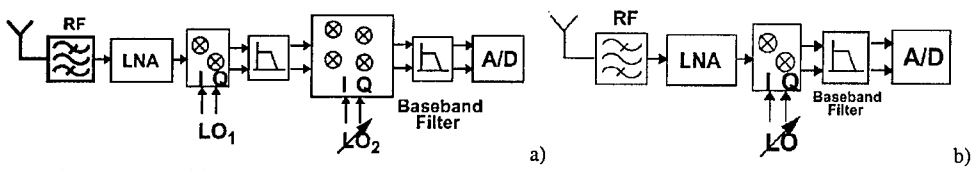
Dążenie do obniżenia poboru mocy w odbiornikach telefonii komórkowej zmusza do poszukiwania nowych rozwiązań układowych. Jednym bloków toru odbiornika GSM jest dolnoprzepustowy filtr pasma podstawowego. W referacie przedstawiono trzy scalone analogowe filtry pasma podstawowego wykonane w technologii CMOS.

1. WPROWADZENIE

Rozwijający się rynek przenośnych urządzeń radiokomunikacyjnych prowadzi do powstawania wielu nowych standardów komunikacyjnych (GSM, DECT, DCS-1800, Bluetooth). Jednocześnie dąży się do zmniejszenia kosztów produkcji, obniżenia pobieranej mocy i zmniejszenia rozmiarów tych urządzeń.

Stąd też obecnie na świecie badania ukierunkowane są na projektowanie w pełni scalonych wielosystemowych odbiorników w najbardziej efektywnych pod względem kosztów technologiach CMOS. Pełna integracja odbiornika w jednym układzie scalonym prowadzi do eliminacji zewnętrznych selektywnych filtrów częstotliwości pośrednich.

Większość komercyjnych urządzeń GSM oparta jest na architekturze odbiornika z podwójną przemianą częstotliwości [1-4]. Dobrą selektywność takiego odbiornika jest uzyskiwana dzięki zastosowaniu dyskretnych filtrów (ceramicznych, SAW, LC). Takie rozwiązanie wymaga jednak stosowania odpowiednich buforów w układzie scalonym, które pobierają znaczną moc do wysterowania niskiej (50Ω) impedancji wejściowej zewnętrznych filtrów. Ponadto przez połączenia zewnętrzne do układu wnoszone są dodatkowe szumy i zakłócenia.



Rys.1. a) Architektura odbiornika z podwójną szerokopasmową przemianą częstotliwości.
b) Architektura odbiornika z bezpośrednią przemianą częstotliwości.

Do pełnej scalonej realizacji najbardziej nadają się odbiorniki z podwójną szerokopasmową (rys.1a) przemianą częstotliwości oraz z bezpośrednią przemianą częstotliwości (rys.2b). Wejściowe pasmo RF zostaje przekonwertowane do częstotliwości pośredniej w pierwszym mieszaczu pracującym ze stałą częstotliwością generatora lokalnego LO1. Niepożądane komponenty sygnału po pierwszej przemianie są usuwane przez prosty filtr dolnoprzepustowy, a następnie sygnały są konwertowane do pasma podstawowego. W odbiorniku z bezpośrednią przemianą częstotliwości [5-6] wejściowe pasmo wielkiej częstotliwości jest bezpośrednio konwertowane do pasma podstawowego. Filtr pasma podstawowego pracuje na małych częstotliwościach i można go zrealizować w postaci scalonej. Dodatkowo przez zapewnienie przestrajania pasma tego filtru możliwa jest praca odbiornika w wielu systemach.

Jednym z ważniejszych bloków funkcjonalnych wymienionych odbiorników, jest filtr pasma podstawowego. Scalone filtry pasma podstawowego mogą być realizowane jako filtry analogowe pracujące z czasem ciągłym (Gm-C, RC), C-przełączane lub cyfrowe. Spośród wymienionych rodzajów filtry cyfrowe i SC gwarantują najlepsze parametry, ale charakteryzują się znacznym poborem mocy i zajmują dużą powierzchnię w układzie scalonym. Analogowe filtry pracujące z czasem ciągłym w nowoczesnych systemów radiokomunikacyjnych znajdują zastosowanie jako filtry antyaliasingowe, natomiast zastosowanie ich do budowy filtrów pasma podstawowego podyktowane jest koniecznością obniżenia poboru mocy i miniaturyzacją przenośnych urządzeń telefonii bezprzewodowej. Z filtrów możliwych do realizacji scalonej, Gm-C wykonane w technologii CMOS cechują się potencjalnie najmniejszym poborem mocy i małą powierzchnia struktury krzemowej.

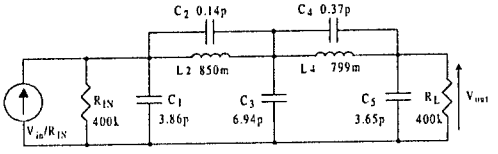
Poniżej zostaną zaprezentowane najważniejsze wyniki i wnioski dotyczące projektów scalonych filtrów pasma podstawowego, które zostały szczegółowo opisane w pracach [7-10].

2. OPIS OPRACOWANYCH FILTRÓW PASMA PODSTAWOWEGO

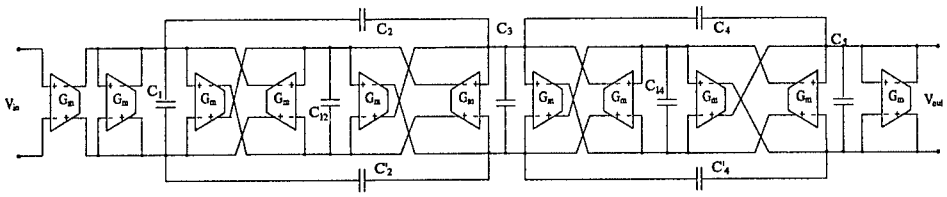
W odbiornikach z bezpośrednią przemianą częstotliwości dla systemów GSM900/1800 filtr pasma podstawowego ma charakterystykę dolnoprzepustową o pasmie przepustowym 100kHz. Realizację tego typu charakterystyki zapewnia dolnoprzepustowy filtr eliptyczny 5-go rzędu, którego pasywny prototyp RLC pokazano na rys. 2.

2.1. Filtry OTA-C

Strukturę aktywnego filtru OTA-C (rys. 3) uzyskano metodą bezpośredniej symulacji prototypu RLC. Filtr zbudowany jest z jedenastu identycznych w pełni różnicowych wzmacniaczy OTA o nominalnej transkonduktancji G_m=2.5μS. W ramach badań wykonano dwa prototypowe filtry scalone, jeden w technologii CMOS 1.5μm oraz drugi w technologii CMOS 0.8μm.



Rys. 2. Prototyp RLC spełniający gabaryty filtru kanałowego GSM.

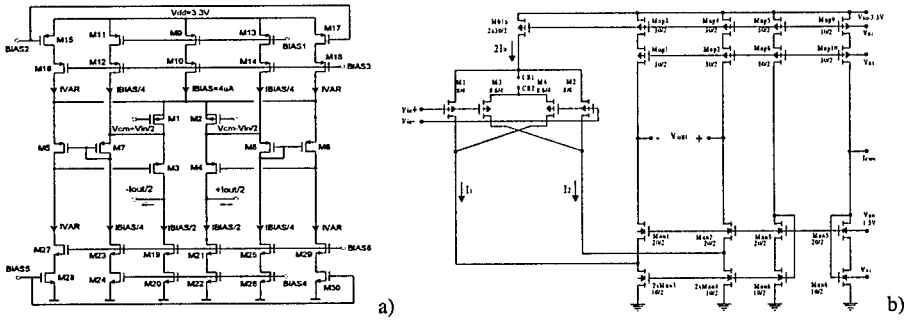


Rys. 3. Realizacja aktywna OTA-C filtru prototypowego.

Do budowy filtru OTA-C w technologii CMOS 1.5μm MOSIS użyto w pełni różnicowe jednostopniowe wzmacniacze OTA (rys.4a) [8]. Liniowy przetwornik napięcie-prąd zastosowany w tych wzmacniaczach zbudowany jest z pary różnicowej z tranzystorami MOS w zakresie triodowym oraz z prostego układu stabilizującego napięcia V_{DS} tych tranzystorów. Wyjścia przetwornika obciążone są kaskodowymi źródłami prądu stałego.

Proces CMOS charakteryzuje się znacznymi odchyłkami parametrów, dlatego istnieje możliwość płynnego elektronicznego przestrajania transkonduktancji wzmacniaczy w zakresie ±30% wokół nominalnej wartości G_m=2.5μS. Taki zakres przestrajania jest wystarczający do kompensacji typowych odchyłek procesu CMOS.

Moc konsumowana oraz powierzchnia zajmowana przez filtr są stosunkowo niewielkie i wynoszą odpowiednio 600μW oraz 0.8mm^2. Relatywnie dużą wartość ma gęstość widmowa szumu wyjściowego (około 4μV/√Hz), głównie z powodu szumu 1/*f*. Wysoki poziom szumu jest cechą charakterystyczną układów CMOS pracujących w zakresie niskich częstotliwości.



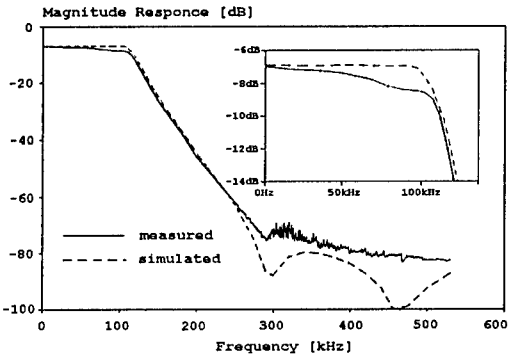
Rys. 4. Wzmacniacze OTA zaprojektowane w technologii CMOS:
a) 1.5μm MOSIS, b) 0.8μm AMS

Filtr OTA-C w technologii 0.8μm AMS wykorzystuje jednostki aktywne OTA o budowie dwustopniowej i w pełni różnicowej (rys. 4b) [9]. Liniowy przetwornik napięcie-prąd stanowiący pierwszy stopień wzmacniacza zbudowany jest z dwóch połączonych krzyżowo par różnicowych MOS pracujących w zakresie nasycenia oraz z wiszącego źródła napięciowego o małej rezystancji, włączonego pomiędzy źródła par różnicowych. Prąd wyjściowy przetwornika napięcie-prąd podawany jest na drugi stopień wzmacniacza utworzony przez kaskodowe źródła prądu stałego.

Wzmacniacze zostały zaprojektowane na nominalną wartość transkonduktancji G_m=2.5μS z możliwością płynnego elektronicznego przestrajania w zakresie ±30%. Cały filtr złożony z jedenastu wzmacniaczy OTA konsumuje moc 2mW przy napięciu zasilania 3.3V. Do korzystnych cech filtru należą małe zniekształcenia nieliniowe wynoszące THD = 0.015% dla napięcia wejściowego o wartości $1V_{pp}$.

Gęstość widmowa szumu wyjściowego jest mniejsza niż w poprzednim filtrze (około 0.7μV/√Hz), dzięki lepszym właściwościom użytej w tym przypadku technologii CMOS.

Ważniejsze parametry filtrów zestawiono w tabeli 1.



Rys. 5. Amplitudowa charakterystyka częstotliwościowa filtru.

Na rys. 5 pokazano pomierzoną charakterystykę amplitudową jednego z prototypowych filtrów. Widać, że w stosunku do charakterystyki symulowanej posiada ona 1dB-we odchylenie w paśmie przepustowym oraz mniejszą wartość tłumienia w paśmie zaporowym. Pomimo występujących rozbieżności charakterystyki prototypowych filtrów mieszczą się w granicach tolerancji filtru pasma podstawowego.

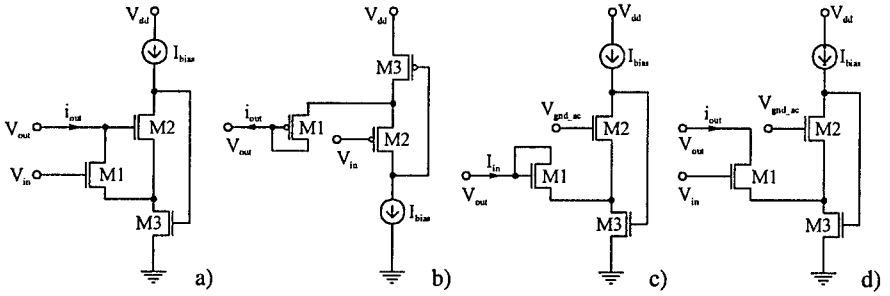
Tabela.1

Podstawowe parametry filtrów

Parametr	Wartość		
Rząd filtru i rodzaj aproksymacji	5, eliptyczny	5, eliptyczny	6, Czebyszewa
Technologia CMOS	MOSIS 1.5μm	AMS 0.8μm	AMS 0.8μm
Napięcie zasilające	3.3V	3.3V	2V
Zakres przestrajania częstotl. granicznej	60-140kHz	56-140 kHz	62-142kHz
THD @ 20 kHz	0.05% @ V_{id}=1V_{pp} 0.2% @ V_{id}=2V_{pp}	0.015% @ V_{id}=1V_{pp} 0.83% @ V_{id}=1.6V_{pp}	1% @V_{id}=186mV_{pp}
Dynamika układu – DR	50.5dB	56dB	54dB
Gęstość widmowa szumu wyjściowego	4μV/√Hz	0.7μV/√Hz	0.6μV/√Hz
Moc pobierana (bez układu strojenia)	0.6mW	2mW	4,7μW
Powierzchnia filtru	0.8 mm^2	0.31mm^2	0.18mm^2

2.2. Filtr Gm-C z kompresją sygnału

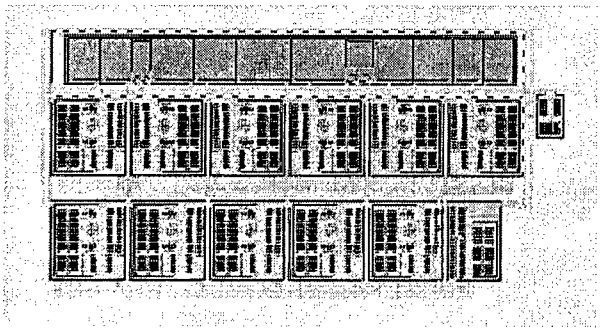
Aby zbadać możliwości zwiększenia dynamiki filtru przy niższym napięciu zasilania został opracowany filtr Gm-C z kompresją sygnału typu *log-domain* [10]. W celu realizacji filtru *log-domain* opracowano podstawowe bloki funkcjonalne CMOS: transkonduktory *log-domain*, kompresor oraz ekspander (rys. 6). Dysponując blokami funkcjonalnymi *log-domain* można zaprojektować filtr aktywny oparty na grafie przepływowym reprezentującym sygnały w pasywnej drabince RLC. Dostrajanie charakterystyk częstotliwościowych filtru możliwe jest dzięki zmianom wartości transkonduktancji integratora *log-domain* uzyskiwanej za pośrednictwem zmian prądów polaryzujących I_{bias}. Opierając się na przedstawionej koncepcji zaprojektowano filtr dolnoprzepustowy 6 rzędu z aproksymacją Czebyszewa.



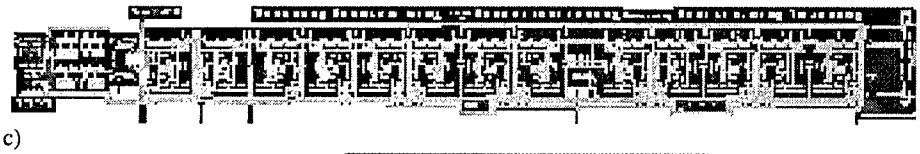
Rys. 6. Podukłady log-domain: a) OTA ujemny, b) OTA dodatni, c) kompresor, d) ekspander.

Symulacje wykazały, że charakterystyki częstotliwościowe filtru z kompresją w nieznacznym stopniu odbiegają od charakterystyk prototypu RLC. Dynamika filtru zdefiniowana została jako stosunek maksymalnej wartości skutecznej sygnału wyjściowego do wartości skutecznej szumu wynosi 54,2dB przy napięciu zasilania 2V. Układ charakteryzuje się bardzo niskim poborem mocy (4,7μW). Zestawienie najważniejszych parametrów filtru podano w tabeli 1. Projekty topografii wszystkich filtrów pokazano na rys. 7.

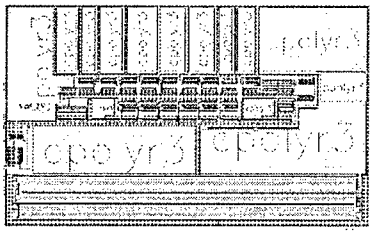
a)



b)



c)



Rys.1. Projekty topografii filtrów a) OTA-C 1.5μm MOSIS (1x0.8mm), b) OTA-C 0.8μm AMS (1.58x0.195mm), c) Log-domain 0.8μm AMS (0.55x0.33mm)

3. PODSUMOWANIE

W pracy przedstawiono trzy realizacje filtrów pasma podstawowego odbiornika GSM z bezpośrednią przemianą częstotliwości lub odbiornika z podwójna szerokopasmową przemianą częstotliwości. Wszystkie opisane filtry mają nominalną częstotliwość graniczną równą 100kHz.

Charakterystyki częstotliwościowe opisanych filtrów spełniają wymagane gabaryty filtru kanałowego. Wszystkie zaprojektowane filtry charakteryzują się możliwością pracy z niskim napięciem zasilania. Opracowane filtry cechują się stosunkowo niskim poborem mocy. Przy obniżonym do 2V napięciu zasilającym szczególnie interesującym rozwiązaniem okazuje się być filtr z kompresją sygnału, w którym dynamikę 54dB uzyskano przy bardzo niskim poborze mocy 4.73μW.

Częstotliwość graniczna opracowanych filtrów może być płynnie przestrajana w zakresie około 60-140kHz. Po zastosowaniu programowalnych matryc kondensatorów częstotliwość graniczną można przestrajać w szerszym zakresie, co umożliwia pracę tych filtrów w scalonych odbiornikach wielosystemowych.

BIBLIOGRAFIA

[1] Iconomos A. Koullias, et al. "A 900 MHz Transceiver Chip Set for Dual-Mode Cellular Radio Mobile Terminals," Digest of Technical Papers, International Solid-State Circuits Conference, pp. 140-141, February 1993

[2] Hisayasu Sato, et al. "A 1.9GHz Single-Chip IF Transceiver for Digital Cordless Phones," Digest of Technical Papers, International Solid-State Circuits Conference, pp. 342-343, February 1996

[3] Chris Marshall, et al. "A 2.7V GSM Transceiver IC with On-Chip Filtering," Digest of Technical Papers, International Solid-State Circuits Conference, pp. 148-149, February 1995

[4] Trudy Stetzler, et al. "A 2.7V to 4.5V Single-Chip GSM Transceiver RF Integrated Circuit," Digest of Technical Papers, International Solid-State Circuits Conference, pp. 150-151, Feb.1995.

[5] Christopher Hull, et al. "A Direct-Conversion Receiver for 900 MHz (ISM Band) Spread-Spectrum Digital Cordless Telephone," Digest of Technical Papers, International Solid-State Circuits Conference, pp. 344-345, February 1996.

[6] Asad Abidi, "Direct-Conversion Radio Transceivers for Digital Communications,"IEEE Journal of Solid State Circuits, Vol. 30, No. 12, pp. 1399-1410, December 1995.

[7] Pankiewicz B., Solecki M.: „Filtr pasma podstawowego odbiornika GSM", Raport badawczy Wydziału ETI PG, nr 18/2002

[8] Jendernalik W.: „Filtr CMOS Gm-C dla przenośnego odbiornika GSM", Raport badawczy Wydziału ETI PG, nr 41/2001.

[9] Jakusz J.: *Realizacja Gm-C analogowego filtru pasma podstawowego odbiornika telefonii komórkowej*, Raport badawczy Wydziału ETI PG, nr 42/2001.

[10] Jendernalik W., Bicki J.: *Projekt analogowego filtru kanałowego CMOS OTA-C z kompresją sygnału dla toru przemiany bezpośredniej odbiornika GSM*, Raport badawczy Wydziału ETI PG, nr 40/2001.

ANALOGUE BASEBAND FILTERS FOR INTEGRATED MOBILE RECEIVERS

Summary

The recent fast developing of battery-operated portable wireless communication devices, especially mobile phones, involves constructing novell low-voltage and low-power analogue circuits. One of that circuit is continuous-time channel-select filter in the front-end of GSM receiver. This paper presents three CMOS implementations of low-power 100kHz low-pass channel-select filters.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Wojciech Kadłubowski

Instytut Elektroniki, Informatyki i Telekomunikacji, Politechnika Szczecińska

STATYSTYCZNA ANALIZA PROCESORÓW ASYNCHRONICZNYCH

Streszczenie

Zaprezentowano wady i zalety zarówno procesorów synchronicznych, jak i asynchronicznych. Przedstawiono model procesora asynchronicznego oraz omówiono statystyczną metodę analizy ich wydajności. Omówiono tworzenie procesu Semi-Markowa opisującego zachowanie się procesora asynchronicznego przetwarzającego instrukcje z danej ścieżki wykonania instrukcji (instruction trace). Na tej podstawie przedstawiono metody porównania różnego typu procesorów.

1. WSTĘP

Rozwój współczesnej telekomunikacji wymaga ciągłego postępu w dziedzinie układów cyfrowych. Zastosowania telekomunikacyjne wymuszają prowadzenie badań w różnych kierunkach. Ostatnio coraz częściej, obok szybkości przetwarzania danych, zwraca się uwagę na zużycie energii oraz na niezawodność działania w różnych warunkach otoczenia. Nikogo nie dziwi chyba fakt, iż takimi właściwościami powinny charakteryzować się np. układy zastosowane w telefonach komórkowych. Od końca lat osiemdziesiątych w wielu ośrodkach badawczych na całym świecie powstają procesory asynchroniczne oraz inne złożone układy nie taktowane zegarem.

W ostatnich latach powstało wiele procesorów asynchronicznych. Można tu wymienić serię procesorów Amulet, a także procesory NSR, Fred, Titac-2 i inne. W większości są to rozwiązania akademickie. Procesor Amulet jest pierwszym komercyjnym rozwiązaniem, które ma swój odpowiednik wśród procesorów synchronicznych (ARM). Został on zaprojektowany z myślą o zastosowaniu w systemach telekomunikacyjnych. W niniejszym artykule zostaną przedstawione wady i zalety rozwiązań asynchronicznych, jak również zaproponowana zostanie statystyczna metoda ich analizy.

1.1. Układy asynchroniczne kontra synchroniczne

Poniżej przedstawione są zarówno wady jak i zalety układów synchronicznych i asynchronicznych.

1.1.1. Układy synchroniczna

Układy synchroniczne, w tym również procesory od wielu lat dominują na rynku układów cyfrowych. Projektanci używają ich ze względu na:

1. *Prostotę projektowania.* Projektowanie jest relatywnie proste w stosunku do układów asynchronicznych. W dużych systemach projektowanie polega na złożeniu modułów komunikujących się przez przesyłanie danych pomiędzy rejestrami taktowanym zegarem.
2. *Spójność systemu.* Zmiany sygnału zegarowego określają kolejne momenty, w których może nastąpić zmiana stanu systemu. Jeśli pominiemy efekt przesunięcia fazy sygnału zegarowego (clock skew), to stan systemu (gdy rozważamy koniec okresu zegara) jest dokładnie określony i dyskretny.
3. *Testowanie.* Łatwiejsze jest również testowanie (techniki testowania są dla nich dobrze rozwinięte).
4. *Powszechność.* Są dobrze znane i rozumiane, a metody projektowania dobrze sformalizowane. Wpływa to na ich szerokie akceptowanie.
5. *Standaryzację.* Standardowe komponenty są szeroko dostępne i dobrze udokumentowane.
6. *Narzędzia projektowe.* Systemy wspomagające projektowanie CAD zostały dobrze rozwinięte wraz z rozwojem techniki synchronicznej.
7. *Weryfikację.* Weryfikacja poprawności operacji jest dość prosta.

Mimo wszystko występują również pewne problemy w projektowaniu systemów synchronicznych:

1. *Problem z zegarem.* Przy dużych częstotliwościach występuje problem z fazą sygnału zegarowego. Powoduje to zwiększone zużycie energii procesora oraz powiększa jego powierzchnię.
2. *Synchronizowanie do najgorszego.* Aby zapewnić spójność przetwarzania danych wszystkie moduły muszą być synchronizowane w takt najwolniejszego cyklu pracy modułu. Ogranicza to szybkość całego systemu.

1.1.2. Układy asynchroniczne

W przypadku układów asynchronicznych często nie można dokładnie przewidzieć opóźnień jego komponentów, a w związku z tym nie można wprowadzić sygnału zegarowego. Taki system może być zaprojektowany jako zbiór modułów funkcjonalnych, z których każdy może pracować ze swoja własną szybkością i współpracujących ze sobą za pomocą odpowiedniej metody komunikacji. Protokół komunikacyjny synchronizuje pracę modułów oraz umożliwia wymianę danych pomiędzy nimi. Najczęściej stosuje się synchronizację sygnałami żądania REQ i potwierdzenia ACK (handshake). Układy asynchroniczne posiadają kilka pozytywnych właściwości w stosunku do ich synchronicznych odpowiedników:

1. *Mniejsze zużycie energii.* Zmiana poziomu sygnału następuje tylko wtedy, gdy jest ona niezbędna. W związku z tym nie ma zużycia energii związanej z sygnałem zegarowym. W niektórych procesorach zużycie energii związane z sygnałem zegarowym wynosi 30– 40%.
2. *Modularność systemu.* Brak sygnału zegarowego doprowadzonego do wszystkich modułów sprawia, iż projektant może koncentrować się na optymalizacji poszczególnych modułów, a następnie bez większych problemów złożyć je w jedną całość. Może to pozwolić na uniknięcie nadmiernej złożoności układu.

3. *Brak problemów z zegarem.* Unikamy problemów związanych z dystrybucją sygnału zegarowego przy wysokich częstotliwościach.
4. *Optymalizacja częstych operacji.* Projektant może skoncentrować się na optymalizacji operacji często występujących. W układach synchronicznych jesteśmy zmuszeni uwzględnić najgorszy przypadek (nawet jeśli rzadko występuje), natomiast w asynchronicznych możemy zadowolić się średnim zachowaniem systemu.
5. *Kompatybilność elektromagnetyczna.* Brak zegara powoduje niższe interferencje elektromagnetyczne. Niskie, nieskorelowane interferencje generowane przez układy asynchroniczne sprawiają, że nadają się one w zastosowaniach, w których użycie systemów synchronicznych jest ograniczone.

Projektowanie układów asynchronicznych ma również pewne wady. Możemy do nich zaliczyć:

1. *Złożoność projektowania.* Złożony proces projektowania oraz brak dobrze opracowanych metod projektowych.
2. *Brak narzędzi.* Brak kompletnych narzędzi CAD dla systemów asynchronicznych.
3. *Testowanie.* Testowanie takich układów nie jest prostym zadaniem.
4. *Szybkość działania.* Nie zawsze można osiągnąć taką szybkość jak w przypadku synchronicznym.
5. *Nadmiarowość powierzchni.* Możliwa nadmiarowość powierzchni układu (np. przy podwójnej szynie).
6. *Zakleszczenia.* Zawsze występuje zwiększone ryzyko występowania zakleszczeń w systemie.

1.2. Metody analizy procesorów synchronicznych i asynchronicznych.

Istnieje wiele metod analizy układów asynchronicznych. Działanie ich jest opisywane za pomocą sieci Petriego lub systemu Event.-Rule[4]. Analizy tego typu są doskonale w przypadku pojedynczych układów. W analizie na poziomie mikroarchitektury procesora wykorzysywane są inne metody. Często tworzy się symulator, a następnie testuje różne konfiguracje procesora. W przypadku procesorów synchronicznych symulator (np. Simple Scalar) wykonuje instrukcje programu w kolejnych cyklach symulacji. Następny stan symulowanego procesora określany jest na podstawie jego aktualnego stanu oraz następnej instrukcji, którą będzie przetwarzana przez procesor. Operacja ta jest powtarzana w każdym cyklu. W procesorach asynchronicznych nie istnieje ustalony cykl procesora. Wszystkie interakcje pomiędzy komponentami procesora następują po wymianie sygnałów synchronizacyjnych (handshake). Różnica pomiędzy symulatorem synchronicznym, a asynchronicznym polega na tym, że stan procesora asynchronicznego jest zmieniany wraz z każdą interakcja pomiędzy komponentami. W obu rodzajach symulatorów (dla procesorów synchronicznych i asynchronicznych) czas symulacji zależy od długości programu testowego i często jest bardzo długi. Symulacje procesora musimy przeprowadzać od nowa dla kolejnych konfiguracji procesora oraz nowych programów testowych. Jest to czasochłonne zadanie i może być nieodzowne w końcowej fazie projektowania procesora. Jednakże w początkowej fazie projektowania, (gdy musimy przebadać dużą przestrzeń możliwych rozwiązań) potrzebna jest inna metoda, która pozwoli zawęzić obszar przeszukiwań. Metoda statystyczna wydaje się być doskonałym rozwiązaniem tego problemu. W metodzie tej ślad ścieżki wykonania programu jest analizowany tylko raz. Tworzone są wtedy prawdopodobieństwa pojawienia się sekwencji instrukcji, a na ich podstawie obliczane są prawdopodobieństwa przejścia do następnego stanu procesora

(tworzony jest dyskretny proces Markowa). Nie obliczamy następnego stanu, do którego przejdzie procesor, lecz prawdopodobieństwo takiego przejścia. Metoda statystyczna dla procesorów synchronicznych typu „in-order" została opisana w [1], a dla procesorów „out-of order" w [6]. W tabeli 1. przedstawiono metody symulacji stosowane zarówno procesorach synchronicznych, jak i asynchronicznych. Poniżej przedstawiono model procesora asynchronicznego oraz omówiona została statystyczna metoda analizy jego wydajności. Aby uwzględnić relacje czasowe pomiędzy komponentami procesora, do opisu jego zachowań użyto proces Semi-Markowa. Wiążąc z poszczególnymi stanami określone cechy procesora (ujawniające się w interakcjach pomiędzy jego komponentami), jesteśmy w stanie analizować jego zachowanie, a także wyznaczać miary jego wydajności. Należy zwrócić uwagę na fakt, iż czas obliczeń związany z metodą statystyczną zależy od przestrzeni stanów, a nie od długości śladu ścieżki wykonania (trace'u). Omawiana metoda dostarcza również informacje, które mogą być wykorzystane w procesie optymalizacji wydajności procesora. Prezentowana metoda może być użyta do analizy procesorów asynchronicznych typu „in-order".

Tablica 1

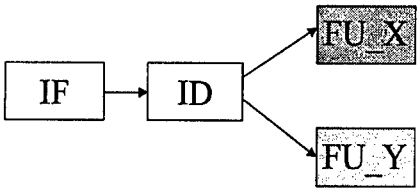
Metody analizy wydajności procesorów

Typ metody	Metoda	Procesory Synchroniczne	Procesory Asynchroniczne
Symulacyjne	Symulatory procesorów (przetwarzające programy wykonywalne).	np.SimpleScalar [3]	np. Symulator procesora Amulet 3 napisany w języku LARD [5]
	Symulatory procesorów (przetwarzające ślad ścieżki wykonania programu)	np. Explorer [2]	np.Symulator wykorzystujący kolorowe sieci Petriego
Statystyczne	Analiza statystyczna	np. Noonburg [1], Smelyanskiy [6]	Prezentowana w tym artykule

2. MODELOWANIE STATYSTYCZNE

2.1. Model procesora

Wszystkie interakcje pomiędzy komponentami zachodzą asynchronicznie. Zakładamy, że przekazywanie instrukcji pomiędzy dwoma komponentami procesora może nastąpić tylko w przypadku, gdy oba są do tego gotowe (nadawca posiada instrukcje gotową do przesłania, a odbiorca jest przygotowany na jej odbiór). Synchronizacja następuje przez wymianę sygnałów Req oraz Ack. Na tym poziomie abstrakcji zakładamy, że opóźnienia związane z sygnałami Ack i Req można pominąć. Opóźnienia te powiązane są z komponentami i reprezentują czas przetwarzania instrukcji. Przyjmujemy, że są one stale. Prosty schemat blokowy procesora przedstawiono na rys. 1. Składa się on z czterech komponentów odpowiadających fazom przetwarzania instrukcji w procesorze Są to: pobranie instrukcji (IF), dekoder instrukcji (ID), jednostka funkcjonalna przetwarzająca instrukcje typu X (FU_X) oraz typu Y (FU_Y)



Rys.1. Model procesora asynchronicznego

Dla przejrzystości rysunku pominięto sygnały Req i Ack. Zakłada się, że FU_X oraz FU_Y nie są nigdy blokowane, tj. instrukcja zawsze może opuścić ostatnią fazę przetwarzania bez dodatkowych opóźnień. Zakłada się również, że zawsze istnieje instrukcja gotowa do rozpoczęcia fazy pobrania instrukcji (tj. pobranie instrukcji po zakończeniu operacji związanych z daną instrukcją nie będzie musiał czekać z rozpoczęciem operacji związanych z następną instrukcją). Przedstawiony model jest dość prosty, jednak dobrze nadaje się do prezentacji statystycznej metody analizy wydajności procesora asynchronicznego.

Przez cykl procesora rozumiemy okres czasu pomiędzy rozpoczęciem fazy pobrania dla dwóch kolejnych instrukcji.

Dla ustalenia uwagi przyjmijmy założenie, że opóźnienia komponentów są różne:

$d(IF) \neq d(ID) \neq d(FU_X) \neq d(FU_Y)$ oraz

$d(IF) = 3$ jednostki czasu, $d(ID) = 2$ jednostki czasu, $d(FU_X) = 5$ jednostek czasu, $d(FU_Y) = 8$ jednostek czasu

Dla powyższych, przykładowych opóźnień zostanie przedstawiony model procesora asynchronicznego. Oczywiście opóźnienia te mogą być dowolne. Należy przy tym pamiętać, że inne opóźnienia będą wpływały na model procesora, w szczególności na fazy przetwarzania instrukcji z_3 i z_4 (patrz definicja stanu procesora).

Niejednakowe opóźnienia poszczególnych komponentów mogą powodować, iż niektóre z nich będą czasami (lub stale) blokowane lub będą „zagłodzone" (starving). Blokowanie między dwoma komponentami następuje w momencie, gdy pierwszy z nich zakończył przetwarzanie instrukcji i jest gotowy do wysłania jej do drugiego, który jest jeszcze w trakcie przetwarzania innej instrukcji. Zagłodzenie jest sytuacją odwrotną. Dany komponent jest gotowy do przyjęcia następnej instrukcji, gdy jego poprzednik nie zakończył jeszcze przetwarzania instrukcji. Łatwo można zauważyć, że komponenty IF oraz ID mogą być blokowane, a ID dodatkowo może czasem być „zagłodzony".

Typy instrukcji

Zakładamy, że w naszym uproszczonym modelu procesor przetwarza dwa typy instrukcji; typu X w jednostce funkcjonalnej FU_X oraz typu Y w FU_Y.

Definicja stanu procesora

Stan procesora opisany jest przez zmienne: $x_1,x_2,x_3,x_4,y_1,y_2,s_1,s_2,z_1,z_2$
gdzie:

- x_i (i =1..4) powiązane są z komponentami procesora IF, ID, FU_X lub FU_Y i oznaczają one ich zajętość lub też stan wolny. Przyjmują one wartość 1 gdy instrukcja znajduje się w danym komponencie lub 0 gdy komponent jest wolny
- y_1,y_2 opisują typ dwóch następnych instrukcji, które jako następne mają opuścić dekoder instrukcji.
- s_i (i=1,2) oznaczają dystans między zależnymi instrukcjami, który jest liczony w liczbie instrukcji. Np. s_1=0 oznacza, że instrukcja będąca w dekoderze jest zależna od

instrukcji, która bezpośrednio przed nią go opuściła, a s_1=1, że jest zależna od drugiej wprzód, która opuściła dekoder. Zmienna s_0 jest przypisana do instrukcji w IF, a zmienna s_1 do instrukcji w ID.

- z_3,z_4 oznacza fazę przetwarzania instrukcji w jednostkach funkcjonalnych FU_X (x_3) lub FU_Y (x_4). Możliwe są następujące wartości:

  start, start n-1,start n-2, starving n-1,starving n-2 lub 0 jeśli odpowiednia wartość x_3 lub x_4 jest równa zero (tzn. jeśli nie ma instrukcji w FU_X lub FU_Y). Powyższe fazy służą obliczeniu czasu trwania poszczególnych stanów procesora.

Definicje faz przetwarzania instrukcji w jednostkach funkcjonalnych:

- start – przetwarzanie instrukcji w jednostce funkcjonalnej rozpoczęło się w cyklu n
- start n-1 – przetwarzanie instrukcji w jednostce funkcjonalnej rozpoczęło się w cyklu n-1
- start n-2 – przetwarzanie instrukcji w jednostce funkcjonalnej rozpoczęło się w cyklu n-2
- starving n-1 – przetwarzanie instrukcji w jednostce funkcjonalnej rozpoczęło się w cyklu n-1 powodując „zagłodzenie" w ID
- starving n-2 – przetwarzanie instrukcji w jednostce funkcjonalnej rozpoczęło się w cyklu n-2 powodując „zagłodzenie" w ID

Zdefiniowane fazy są dopasowane do przykładowych opóźnień komponentów procesora, tj:

d(IF) = 3 jednostki czasu, d(ID) = 2 jednostki czasu, d(FU_X) = 5 jednostek czasu, d(FU_Y) = 8 jednostek czasu

Zmienione opóźnienia komponentów oraz ich różna ilość mogą wpływać na model procesora asynchronicznego, a w szczególności definicje faz. Procedura definiowania faz dla danych opóźnień komponentów została pominięta ze względu na brak miejsca.

2.2. Tworzenie procesu Semi-Markowa

Metoda statystyczna może być wykorzystana do obliczenia miar związanych z procesorami asynchronicznymi. Należą do nich np. cykl średni układu, współczynnik wykorzystania poszczególnych komponentów systemu, czas wykonania programu.

Najpierw analizowana jest ścieżka wykonania programu i określane są prawdopodobieństwa wystąpienia sekwencji instrukcji. Następnie, w oparciu o model procesora asynchronicznego (zdefiniowanego w punkcie 2.1) oraz o prawdopodobieństwa wystąpienia sekwencji instrukcji, konstruowany jest proces Semi-Markowa. Opisuje on zachowanie się procesora asynchronicznego przetwarzającego instrukcje z danego śladu ścieżki wykonania programu. Po skonstruowaniu procesu Semi-Markowa, możemy określić dla niego prawdopodobieństwa graniczne jego stanów, jak również czasy ich trwania. Na ich podstawie jesteśmy w stanie policzyć interesujące nas miary. Przykładowo możemy powiązać blokowanie danego komponentu z konkretnymi stanami procesora, a następnie obliczyć dla nich prawdopodobieństwa graniczne. W ten sposób możemy obliczyć jak często następuje blokowanie komponentu dla danego śladu ścieżki wykonania programu.

Procedura obliczania miar wydajności jest dwuczęściowa. Najpierw tworzymy macierze **P** i **S** określające proces Semi-Markowa (**P** jest macierzą prawdopodobieństw przejść procesu wbudowanego, a macierz **S** jest macierzą prawdopodobieństw pobytu w jego stanach), a następnie na ich podstawie obliczamy interesujące nas miary.

Tworzenie macierzy **P** i **S**:

1. Dla każdego stanu procesora znajdujemy odpowiadającą mu sekwencję instrukcji w śladzie ścieżki wykonania programu (np. sekwencję 5 instrukcji)
2. Dla każdej sekwencji obliczamy prawdopodobieństwo pojawienia się po niej instrukcji X lub Y (zakładając, że procesor przetwarza dwa typy instrukcji – patrz model procesora). Odpowiada to znalezieniu prawdopodobieństwa przejścia procesora z jednego stanu do następnego. Na przykład, załóżmy, że sekwencji seq1 odpowiada stan1. Załóżmy również, że po sekwencji seq1 pojawia się instrukcja typu X z prawdopodobieństwem 0.3, a instrukcja Y z prawdopodobieństwem 0.7. Odpowiada to przejściu procesora ze stanu 1 do stanu 2 (instrukcja X) z prawdopodobieństwem 0.3, a do stanu 3 (instrukcja Y) z prawdopodobieństwem 0.7.
3. Dla każdego stanu znajdujemy czas jego trwania (tj. czas pobytu procesora w danym stanie)
4. Dla tak stworzonego procesu Semi-Markowa obliczamy prawdopodobieństwa graniczne znalezienia procesu (a co za tym idzie i procesora) w danym stanie.

Obliczanie metryk:

1. Identyfikujemy interesujące nas stany procesora (np. stany, w których następuje blokowanie danego komponentu procesora)
2. Obliczamy prawdopodobieństwo ich wystąpienia
3. Obliczamy miary wydajności (np. mając czasy pobytu procesora w poszczególnych stanach oraz prawdopodobieństwo ich wystąpienia możemy policzyć średni cykl procesora asynchronicznego; możemy również obliczyć zajętość poszczególnych komponentów procesora)

3. ANALIZA PORÓWNAWCZA

Porównanie wydajności procesorów osiągamy przez uruchamianie na nich programów testujących i obliczanie odpowiednich miar wydajności. Wykorzystywane są w tym celu rozbudowane programy testowe (benchmark) SPEC lub też proste programy syntetyczne (np. Dhrystone). Użycie metod statystycznych umożliwia porównanie wydajności zarówno procesorów synchronicznych, jak również procesorów asynchronicznych. Należy tylko zastosować metodę odpowiednią do danego typu procesora (np. [1], [6], lub prezentowaną w tym artykule). Porównanie można dokonywać w trzech różnych klasach procesorów:

- synchroniczne – synchroniczne
- synchroniczne – asynchroniczne
- asynchroniczne – asynchroniczne

Dla porównania procesorów synchronicznych najczęściej wykorzystywaną miarą jest średnia ilość instrukcji przetwarzana w ciągu jednego cyklu procesora (IPC – Instruction per Cycle). Taka syntetyczna miara obrazuje cechy mikroarchitektury procesorów synchronicznych. Oczywiście czas wykonania programu jest najbardziej wiarygodną miarą ich wydajności. W dziedzinie procesorów asynchronicznych często wykorzystywanymi miarami są: średni cykl procesora dla danego programu testowego oraz zajętość poszczególnych komponentów procesora. Informują one o cechach mikroarchitektury (np. o „wąskich gardłach" w systemie). Również i w tym przypadku możemy badać czas wykonania programu. Procesory synchroniczne z asynchronicznymi możemy porównać używając wspólnej miary dla obu typów procesorów; tj. czasu wykonania programu.

W tablicy 2 przedstawiono porównanie dokładności metody statystycznej w stosunku do symulacyjnej. Porównanie przeprowadzono używając programu testowego 1 (a właściwie jego śladu ścieżki wykonania instrukcji programu), stworzonego specjalnie w tym celu. Program testowy został stworzony tak, by wyeksponować lokowania i zagłodzenie dekodera instrukcji. Jak widać różnica pomiędzy metodą symulacyjną, a statystyczną jest bardzo mała.

Tablica 2

Dokładność metody statystycznej

	Symulacja (cykl średni)	Metoda statystyczna (cykl średni)
Program testujący 1	4.857	5.147

4. WNIOSKI

Prezentowana metoda umożliwia analizę wydajności procesorów asynchronicznych oraz analizy ich „wąskich gardeł". Za jej pomocą projektant jest w stanie analizować przestrzeń stanów procesora we wstępnej fazie projektowania. Umożliwia to podejmowanie decyzji projektowych bez konieczności wykonywania długotrwałych symulacji. Dzięki temu można dobrać optymalne opóźnienia dla danego komponentu oraz wskazać ścieżeki krytyczne, które należy minimalizować. Metoda statystyczna dostarcza również dane, które mogą być wykorzystane przez kompilator do optymalizacji kodu, w tym do zminimalizowania blokowaniń pomiędzy komponentami procesora.

BIBLIOGRAFIA

[1] D.N. Noonburg, J.P. Shen A *framework for statistical modeling of superscalar processor performance*,
[2] T.D. Diep, J.P. Shen, M. Phillip, *EXPROLER: A Retargetable and visualsation-based trace driven simulator*, Proc. Micro-26, str.225-235, 1993
[3] D. Burger, T.M. Austin, *The SimpleScalar Tol Set, Version 2.0*,Computer Architecture News, 25(3):13-25,1997
[4] S.M. Burns, A.J. Martin, *Performance analysis and optimization of asynchronous circuits*, in Advanced Research in VLSI Conference, Santa Cruse, CA, March 1991
[5] P. Endecott, S. Furber, *Modelling and simulation of asynchronous systems using the LARD hardware description language.*
[6] M.Smelyanskiy, E.S. Davidson, D. Burger, *Fast and accurate performance modeling of out-of-order issue processors.*

THE STATISTICAL METHOD FOR ANALYSIS OF ASYNCHRONOUS PROCESSORS

Summary

Advantages and disadvantages of synchronous and asynchronous processors are presented. The model of asynchronous processor was discussed and the statistical method of performance analysis was shown. The base of analysis is a Semi-Markov process, which described a behavior of asynchronous processor. Statistical method allows a comparison of synchronous and asynchronous processors.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Miron Kłosowski, Sławomir Malinowski

Katedra Systemów Mikroelektronicznych, Politechnika Gdańska

IMPLEMENTACJA WYBRANYCH STRUKTUR SZTUCZNYCH SIECI NEURONOWYCH W CYFROWYCH UKŁADACH PROGRAMOWALNYCH

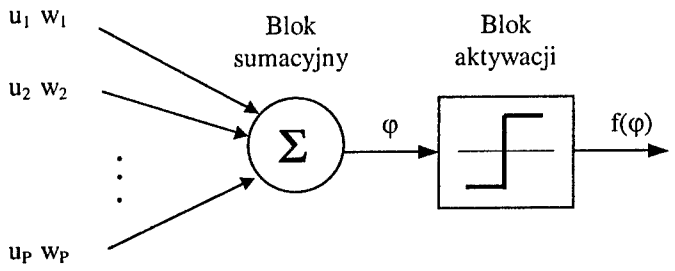
Streszczenie

W pracy przedstawiono zagadnienia związane z budową i implementacją sztucznych sieci neuronowych w układach programowalnych typu FPGA. Szczegółowo omówiono implementację pojedynczego neuronu z wykorzystaniem dostępnych zasobów sprzętowych układu Virtex FPGA. Poruszono również zagadnienie optymalizacji struktury sieci do konkretnych zastosowań i możliwości układów FPGA. Zdefiniowano trzy rodzaje realizacji neuronu: równoległą, równoległo-sekwencyjną i sekwencyjną. Z wykorzystaniem wyżej wymienionych realizacji zbudowano sieci składające się z kilkudziesięciu neuronów. Sieci te zostały przetestowane w prostych zastosowaniach takich jak np.: rozpoznawanie cech. W pracy podsumowano rezultaty realizacji sieci w układach programowalnych FPGA takie jak: zajętość układu FPGA w funkcji rozmiarów sieci, częstotliwość zegarowa, liczba wejść i wyjść układu.

1. WSTĘP

Tematem pracy jest implementacja struktury sztucznej sieci neuronowej [1] w układzie programowalnym. Proces przetwarzania informacji przez sieć neuronową oraz jego efekt silnie związany jest ze strukturą sieci. Nie każde zagadnienie rozwiązywane jest za pomocą sieci o takiej samej i niezmiennej strukturze. W zależności od realizowanego problemu istnieje konieczność doboru właściwej struktury. Większość zrealizowanych sieci istnieje tylko w postaci programu komputerowego lub analogowego układu elektronicznego. Jakość pracy sieci stworzonej fizycznie zależy również od użytych do jej budowy elementów. Najprostsze neurony zbudowane są na wzmacniaczach operacyjnych jako sumatory ze wzmocnieniem określonym przez dołączone zewnętrzne rezystory. Rezystory te pełnią rolę wag oraz ustalają funkcję aktywacji. Uczenie takiej sieci jest bardzo trudne ze względu na konieczność zmian wartości rezystancji. W poniższej pracy przedstawiono propozycję stworzenia cyfrowej sieci neuronowej zaimplementowanej w układzie programowalnym. Rozwiązanie to przyspiesza proces zmiany struktury sieci w zależności od potrzeb i problemu do rozwiązania. W pracy przedstawiono sposób budowy i implementacji neuronu oraz całej sieci typu feed-forward [2] z wykorzystaniem programu Active-HDL i języka VHDL. Przedstawione zostały także zadania przeznaczone do

rozwiązania przez zaprojektowaną sieć neuronową oraz wyniki przeprowadzonych eksperymentów i pomiarów działających struktur sieci neuronowych spełniających wcześniej zdefiniowane zadania. Pomiarów i testów dokonywano programem ISE 5.2i oraz na płytce prototypowej z układem programowalnym Virtex. Stworzony program uczenia sieci generuje w wyniku pliki formatu .vhd. Przedstawiono także budowę pojedynczego neuronu i sposoby implementacji w układzie programowalnym całej struktury sztucznej sieci neuronowej wykorzystując do opisu język VHDL. Na rys. 1 przedstawiono model neuronu [3] wykorzystywany do implementacji.



Rys. 1. Budowa neuronu

$$\varphi = \sum_{i=1}^{P} u_i w_i \tag{1.1}$$

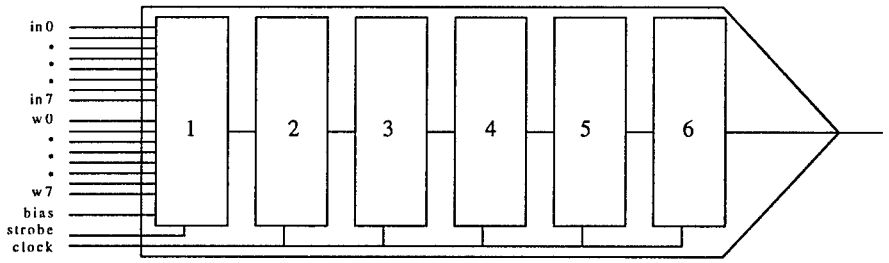
gdzie:
u_i – wartość i-tego sygnału wejściowego,
w_i – wartość i-tej wagi,
P – liczba wejść,
φ – potencjał membranowy.

$$f(\varphi) = \frac{1}{1+e^{-\beta\varphi}} \tag{1.2}$$

gdzie:
β – zadany parametr.

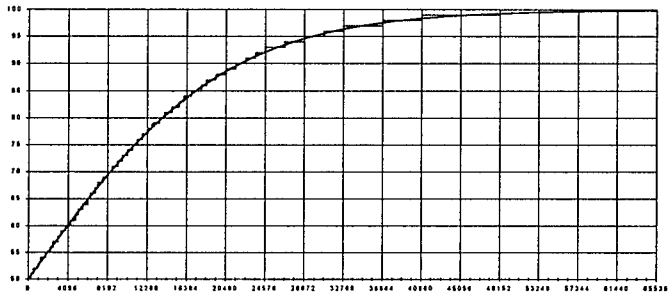
2. BUDOWA NEURONU

W związku z trudną implementacją arytmometrów zmiennoprzecinkowych w układach programowalnych [5] zdecydowano o wyborze stałoprzecinkowej reprezentacji sygnałów pobudzeń i wyjść neuronów. Wielkości te są skalowane przez 100, co umożliwia przedstawienie ich z dokładnością do dwóch miejsc po przecinku w postaci integer. Przeskalowanie wag odbywa się w programie uczącym. Na wejścia neuronu są podawane wielkości przeskalowane. Dla zwiększenia szybkości działania neuron pracuje na obu zboczach zegara. Schemat blokowy pojedynczego neuronu przedstawiono na rys. 2.



Rys. 2. Implementacja pojedynczego neuronu.

1. Przy narastającym zboczu sygnału strobe obliczany jest potencjał membranowy φ. W procesie syntezy i implementacji mnożenie jest realizowane w dostępnych mnożnikach sprzętowych, a jeżeli operacji mnożenia jest więcej to realizowane są one w logice układu programowalnego.
2. Przy opadającym zboczu zegara sprawdzany jest znak sygnału φ.
3. Przy narastającym zboczu zegara sprawdzana jest wielkość liczby φ, czy mieści się w dziedzinie funkcji aktywacji oraz jej konwersja z postaci integer na postać binarną.
4. Przy opadającym zboczu zegara realizowana jest konwersja fragmentów sygnału φ z postaci binarnej na postać integer.
5. Przy narastającym zboczu obliczana jest wartość funkcji aktywacji dla danej liczby φ na podstawie kilku bitów znaczących (ich liczba zależy od wielkości liczby φ). Aproksymację funkcji aktywacji przedstawiono na rys. 3.
6. Przy zboczu opadającym na wyjście jest podawana odpowiedź neuronu uzależniona od wartości funkcji aktywacji oraz znaku liczby φ.



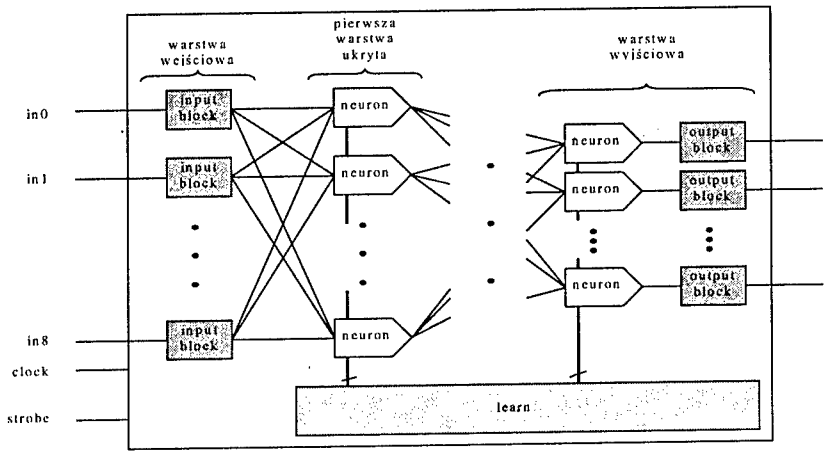
Rys. 3. Aproksymacja funkcji aktywacji neuronu.

3. BUDOWA SIECI NEURONOWEJ

Sieć neuronowa jest zrealizowana jako połączenie warstw neuronów z pewnymi układami pośredniczącymi (jak pamięć wag, układy wejścia i wyjścia). Odpowiedzialny za to jest plik network.vhd tworzony przez program uczący. Musi on związać w jedną całość pliki:

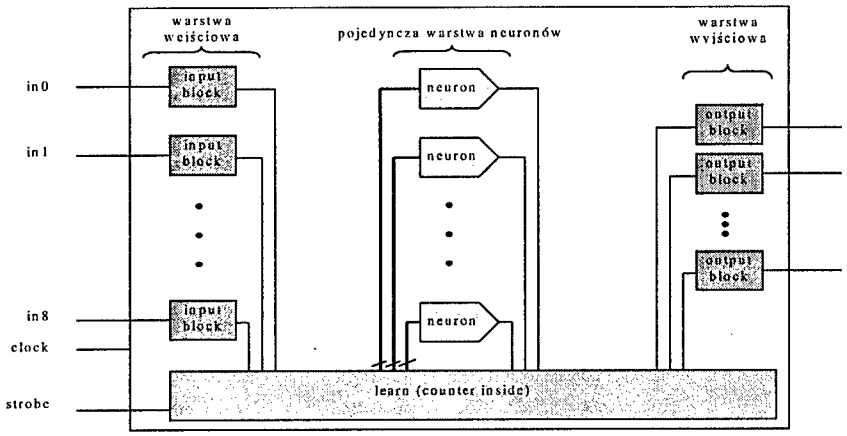
1. neuron.vhd – pojedynczy neuron;
2. learn.vhd – blok zawierający wagi oraz sterujący siecią;
3. input_block – bufor wejściowy skalujący;
4. output_block – blok wyjściowy progowy.

Zadaniem dwóch ostatnich jest dopasowanie cyfrowych sygnałów zewnętrznych do sygnałów typu integer używanych w układzie. Przeskalowanie sygnału wejściowego podawanego na układ odbywa się w warstwie wejściowej zbudowanej z bloków input_block w ten sposób, że logicznemu 0 odpowiada liczba 0 a logicznej 1 odpowiada liczba 100. Za zamianę sygnału wyjściowego z postaci integer do cyfrowej odpowiada układ output_block. Ustalona w nim wartość progowa decyduje o wartości logicznej na wyjściu sieci. Domyślnie przyjęto wartość 80. Oznacza to, że przy odpowiedzi neuronu większej lub równej 80 na wyjściu otrzymamy 1, jeżeli jednak odpowiedź będzie mniejsza od 80 to otrzymamy 0. W zależności od wyboru dokonanego przez użytkownika, sieć może przyjąć trzy różne konfiguracje, które nie tylko różnią się budową, ale również przewidywanymi prędkościami dokonywanych obliczeń oraz budową bloku learn.vhd, generowanego przez program uczący. Pierwszą z możliwych implementacji przedstawia rys. 4.



Rys. 4. Implementacja równoległa sieci neuronowej.

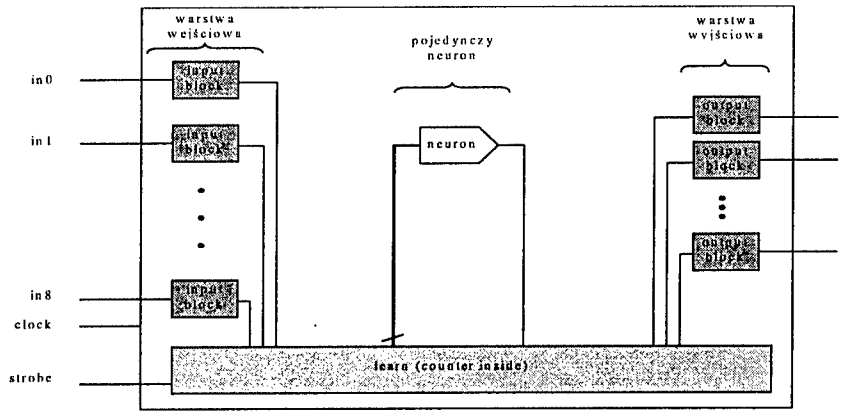
W tym przypadku implementowana jest cała sieć neuronowa. Przy takiej budowie sieci można przewidzieć, że wynik na wyjściu zostanie osiągnięty po czasie, jaki potrzebny jest na przetworzenie danych przez jedną warstwę zwielokrotniony przez liczbę warstw (nie licząc wejściowej). Przy taktowaniu zegarem o częstotliwości 40MHz wynik na wyjściu sieci z trzema warstwami ukrytymi otrzymamy po około 4*75ns = 300ns. Jest to czas bardzo mały, jednak takie rozwiązanie nie jest ekonomiczne pod względem zajętości powierzchni układu programowalnego. Przy dużej liczbie neuronów, część ich wewnętrznych układów mnożących jest realizowana na logice układu, a nie na mnożnikach sprzętowych, co dodatkowo zwiększa zajętość powierzchni układu programowalnego. W realizacji równoległej układ learn zawiera tylko wartości wag neuronów. Neurony sterowane są tylko zegarem układu, który jest również podłączony do wejścia strobe. W ten sposób zaimplementowana sieć neuronowa ma istotną zaletę - jest w stanie przetwarzać sygnały wejściowe z każdym cyklem zegarowym, czyli działa jako potok. Gdy sygnał wejściowy zmienia się z częstotliwością taktowania lub mniejszą żadne dane nie zostaną utracone. Na rys. 5 przedstawiono kolejną implementację. W tym przypadku implementowana jest tylko jedna warstwa neuronów, która później jest przełączana pomiędzy poszczególnymi rejestrami.



Rys. 5. Implementacja równoległo-sekwencyjna.

Przy takiej budowie sieci można przewidzieć, że wynik na wyjściu zostanie otrzymany po czasie zbliżonym do takiego jaki jest w realizacji poprzedniej. To rozwiązanie jest bardziej ekonomiczne pod względem zajętości powierzchni układu programowalnego aniżeli rozwiązanie poprzednie. Większe układy Virtex posiadają już wystarczającą liczbę mnożników sprzętowych aby wszystkie 72 operacje mnożenia potrzebne do realizacji sieci neuronowej nie wymagały użycia logiki programowalnej do tworzenia kolejnych mnożników. W realizacji równoległo-sekwencyjnej układ learn zawiera oprócz wartości wag neuronów także generator wewnętrznego sygnału strobe sterującego neuronami oraz licznik odpowiedzialny za prawidłową pracę całej sieci. Układ ten zawiera również rejestry, które są odpowiedzialne za przechowywanie danych wejściowych oraz wyników obliczeń kolejnych warstw. Na podstawie zgromadzonych wyników otrzymujemy wynik przetworzenia sygnału przez całą sieć. Istotną różnicą w tej realizacji jest brak zdolności przetwarzania sygnałów z tak dużą szybkością jak w realizacji równoległej, ponieważ układ tan nie posiada zdolności przetwarzania potokowego. Oznacza to, że sygnał wejściowy musi się zmieniać z mniejszą częstotliwością, która dla sieci z 3 warstwami ukrytymi wynosi około 1/300ns = 3,33MHz.

Trzeci sposób implementacji przedstawia Rys. 6. W tym przypadku implementowany jest tylko jeden neuron, który później jest przełączany pomiędzy poszczególnymi rejestrami. Przy takiej budowie sieci można przewidzieć, że wynik na wyjściu otrzymamy po bardzo długim czasie wynikającym z liczby neuronów oraz warstw w sieci. Taką zasadę działania można porównać z softwarową wersją sieci, gdzie wszystkie obliczenia odbywają się w pętlach, neuron po neuronie. Rozwiązanie to jest najbardziej ekonomiczne pod względem zajętości powierzchni układu programowalnego [6]. Wykorzystuje tylko 9 mnożników sprzętowych. W realizacji szeregowej układ learn tak jak poprzednio zawiera oprócz wartości wag neuronów także generator wewnętrznego sygnału strobe sterującego neuronami oraz licznik odpowiedzialny za prawidłową pracę całej sieci. Podobnie układ ten zawiera również rejestry, które są odpowiedzialne za przechowywanie danych wejściowych oraz wyników obliczeń kolejnych neuronów, na podstawie których w rezultacie otrzymujemy wynik przetworzenia sygnału przez całą sieć. Także w tej realizacji nie ma możliwości przetwarzania sygnałów z tak dużą szybkością, ponieważ układ ten nie posiada zdolności przetwarzania potokowego. Sygnał wejściowy musi się zmieniać z bardzo małą częstotliwością, która jest zależna od liczby neuronów w poszczególnych warstwach.



Rys. 6. Implementacja sekwencyjna.

Wybór konkretnej realizacji jest uzależniony od rodzaju sieci jaką chcemy zbudować i od zadania jakie ona ma wykonywać. Pierwsza z nich doskonale nadaje się do realizacji małych sieci o dużej prędkości działania i koniecznością stosowania w systemach wymagających dużej odporności na utratę danych. Druga z nich jest kompromisem pod względem prędkości działania oraz zajętości powierzchni układu programowalnego. Dobrze sprawdzi się na przykład w układach przetwarzania dźwięku, gdzie dołączone zewnętrzne przetworniki A/C mają czas ustalania rzędu kilku lub kilkudziesięciu ms. Dźwięk wysokiej jakości jest próbkowany z częstotliwością 44100Hz, co jest znacznie mniejsze od częstotliwości pracy równoległo–szeregowej implementacji sieci neuronowej. Można więc założyć, że ten rodzaj układu dobrze sprawdzi się w roli różnych rodzajów filtrów i przetworników dźwięku. Trzecia z nich ma zasadę pracy identyczną z programową realizacją. Jej wykorzystanie w układach fizycznych jest nie opłacalne, gdyż taki układ można zastąpić systemem mikroprocesorowym. Ta realizacja jest przedstawiona dla celów porównawczych.

4. REZULTATY IMPLEMENTACJI

Jako przykład zrealizowano sieć neuronową do rozpoznawania cyfry na wyświetlaczu 7-segmentowym. Do ręcznego testowania sieci wykorzystuje się dostępny na płycie prototypowej układu Virtex przełącznik DIP, którym ustawia się kształt cyfry do rozpoznania. Ustawiona cyfra wyświetlana jest na lewym wyświetlaczu. Rezultat rozpoznawania wyświetlany jest na prawym wyświetlaczu. Można dzięki temu obserwować działanie sieci, gdy podana informacja wejściowa jest niepełna. Przykład działania przedstawiono w tablicy 4.1. W realizacji równoległej wynik zmienia się bezpośrednio po zmianie pozycji któregokolwiek z przełączników. W realizacjach równoległo-sekwencyjnej i sekwencyjnej wynik otrzymamy dopiero po naciśnięciu przycisku strobe. Zrealizowane układy korzystały z dwóch dostępnych na płytce zegarów. Zegar o częstotliwości taktowania 24 MHz używany był do sterowania pracą sieci neuronowej, natomiast zegar o częstotliwości 100 MHz używano do taktowania bloku dekodera odpowiedzi sieci na sygnał wyświetlany przez wyświetlacz 7-segmentowy.

Tablica 4.1

Działanie sieci o różnych strukturach w warunkach niepewności.

Symbol do rozpoznania	Struktura sieci (liczba neuronów w kolejnych warstwach) i rozpoznana cyfra					
	7-10	7-8-10	7-7-10	7-6-10	7-5-10	7-4-10
	0	0	0	0	8	3
	2	2	2	2	2	2
	5	5	5	5	5	6
	9	8	9	9	9	8

Tablica 4.2

Parametry implementacji rozpoznawania znaku na wyświetlaczu 7-segmentowym.

parametr czasowy	rodzaj realizacji		
	równoległa	równoległo – sekwencyjna	sekwencyjna
minimum period (ns)	32.994	17.620	18.466
maximum frequency (MHz)	30.309	56.754	54.154
minimum input arrival time before clock (ns)	16.559	6.959	6.122
maximum output required time after clock (ns)	10.399	9.114	13.035

Tablica 4.3

Zajętość układu programowalnego do realizacji rozpoznawania znaku na wyświetlaczu 7-segmentowym.

rodzaj elementu	dostępne	rodzaj realizacji		
		równoległa	równoległo - sekwencyjna	sekwencyjna
total number of Slice Registers	10240	1438	1586	475
number of occupied Slices	5120	6069	5312	839
total number 4 input LUTs	10240	10168	10171	1358
number of bonded IOBs	324	23	24	24
number of MULT18X18s	40	-	-	8
number of GCLKs	16	2	2	2
total equivalent gate count for design	1000000	115963	187697	110779

W tablicy 4.3 przedstawiono porównanie realizacji pod względem wykorzystania powierzchni układu programowalnego. Nie wszystkie realizacje dały się zaimplementować. Realizacje równoległa i równoległo-sekwencyjna zajęły więcej powierzchni niż było dostępne w układzie 2V1000. Realizacja sekwencyjna to jedyna, która zmieściła się

w użytym układzie programowalnym. Działa zgodnie z przeprowadzoną wcześniej symulacją. Po implementacji otrzymano informacje z jakimi częstotliwościami maksymalnymi może pracować układ (przedstawiono to w tablicy 4.2).

5. ZAKOŃCZENIE

W pracy przedstawiono budowę pojedynczego neuronu oraz realizacje równoległą, równoległo-sekwencyjną i sekwencyjną sieci neuronowej. Jako przykład implementacji podano zadanie rozpoznawania znaku na wyświetlaczu siedmiosegmentowym. Podano rezultaty implementacji takie jak: zajętość układu FPGA w funkcji rozmiarów sieci, częstotliwość zegarowa, liczba wejść i wyjść układu. Zaimplementowany układ został przetestowany i działał zgodnie z wcześniej dokonanymi symulacjami. Dalsze badania dotyczące opisanego zagadnienia będą obejmowały:

- wykorzystanie do implementacji większych układów FPGA niż obecnie używany VirtexII 2V1000 co umożliwi realizację równoległą nawet bardziej złożonych sieci;
- zmianę sposobu realizacji obliczania potencjału membranowego φ oraz aproksymacji funkcji aktywacji celem zmniejszenia zajmowanej powierzchni przez układ;
- implementację innych typów sieci neuronowych;
- podłączenie zewnętrznych układów, np. procesora, umożliwiających dynamiczne zmiany wag neuronów w czasie pracy;
- implementację w układzie FPGA algorytmów trenujących sieć neuronową.

BIBLIOGRAFIA

[1] Białko M.: *Podstawowe właściwości sieci neuronowych i hybrydowych systemów ekspertowych.* Wydawnictwo Uczelniane Politechniki Koszalińskiej, Koszalin 2000.
[2] Osowski S.: *Sieci neuronowe w ujęciu algorytmicznym.* Wydawnictwo Naukowo Techniczne, Warszawa 1996.
[3] Rojas R.: *Neural Networks – A Systematic Introduction.* Springer-Verlag, Berlin, New-York 1996.
[4] Żurada J., Barski M., Jędruch W.: *Sztuczne sieci neuronowe.* PWN, Warszawa 1996.
[5] Armstrong J. R., Gray F. G.: *VHDL Design Representation and Synthesis.* Prentice Hall PTR, New Jersey 2000.
[6] Zwoliński M.: *Projektowanie układów cyfrowych z wykorzystaniem języka VHDL.* Wydawnictwa Komunikacji i Łączności, Warszawa 2002.

IMPLEMENTATION OF SOME ARTIFICIAL NEURAL NETWORK ARCHITECTURES IN DIGITAL PROGRAMMABLE CIRCUITS

Summary

In the paper issues concerning structure and FPGA IC implementation of artificial neural networks are presented. Neuron implementation using Virtex FPGA has been presented in details. Optimization of neural network structure for particular application has been discussed. Three types of neural network implementation are presented: parallel, parallel-sequential and sequential. Artificial neural networks of up to 100 neurons can be implemented in large FPGA this way. Simple implementation example of neural network for pattern recognition has been presented. Network recognizes patterns on 7-segment display. Implementation results like FPGA resource usage, clock frequency, number of input and output blocks are described.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Wiesław Kordalski[*], Michał Polowczyk[*], Janusz Czerwiński[+], Jerzy Dzierżko[+], Lech Dobrzański[#]

[*] Katedra Systemów Mikroelektronicznych, Politechnika Gdańska
[+] Commonet, Gdańsk
[#] Instytut Technologii Materiałów Elektronicznych, Warszawa

NOWY, DWUDRENOWY TRANZYSTOR MOS JAKO CZUJNIK POLA MAGNETYCZNEGO

Streszczenie

W artykule zaprezentowano nową koncepcję tranzystora typu MOS z dwoma poziomo usytuowanymi drenami, który może być wykorzystany jako czujnik pola magnetycznego. Zasada działania nowego czujnika oparta jest na jednym ze zjawisk galwanomagnetycznych. Na podstawie wyników pomiarowych dostępnych w literaturze przedmiotu można wykazać, że przewidywana czułość proponowanego czujnika może być ok. 1000 razy większa niż czułość znanych i cenionych czujników wykorzystujących tranzystory MAGFET. Czujnik według nowej konstrukcji ma niezwykle małe wymiary i absorbuje małą moc.
Artykuł składa się z krótkiego wstępu, opisu zasady działania nowego czujnika, oszacowania wrażliwości czujnika na pole magnetyczne oraz konkluzji.

1. WSTĘP

Czujnik pola magnetycznego jest elementem wejściowym przetwornika dokonującego konwersji pola magnetycznego $\boldsymbol{H}$ na użyteczny sygnał elektryczny i może być stosowany zarówno do bezpośrednich pomiarów i detekcji pola magnetycznego, jak i do pomiarów i detekcji wielkości niemagnetycznych, na przykład pomiaru prądu za pośrednictwem pola magnetycznego przezeń wytworzonego lub pomiaru przemieszczenia magnesu. W klasie zastosowań bezpośrednich, czujnik pola magnetycznego jest częścią magnetometrów służących do pomiaru pola magnetycznego ziemi, czytania taśm i dysków magnetycznych, rozpoznania banknotów i wszelkiego rodzaju kart z nośnikami magnetycznymi oraz wykorzystywany jest do sterowania aparaturą magnetyczną. Z kolei, w klasie zastosowań pośrednich, czujnik ten wykorzystywany jest do zdalnego przełączania kluczy, detekcji i pomiaru przemieszczeń liniowych i kątowych, nieinwazyjnego pomiaru mocy elektrycznej, etc. Bardzo ważnym parametrem magnetometrów jest ich czułość na wartość i zmianę pola magnetycznego. Trwają intensywne prace nad nowymi rozwiązaniami czujników o prostej konstrukcji i dużej czułości (zakres nanotesli), które zastąpiłyby kosztowne i duże magnetometry, których zasada działania oparta jest na magnetycznym rezonansie

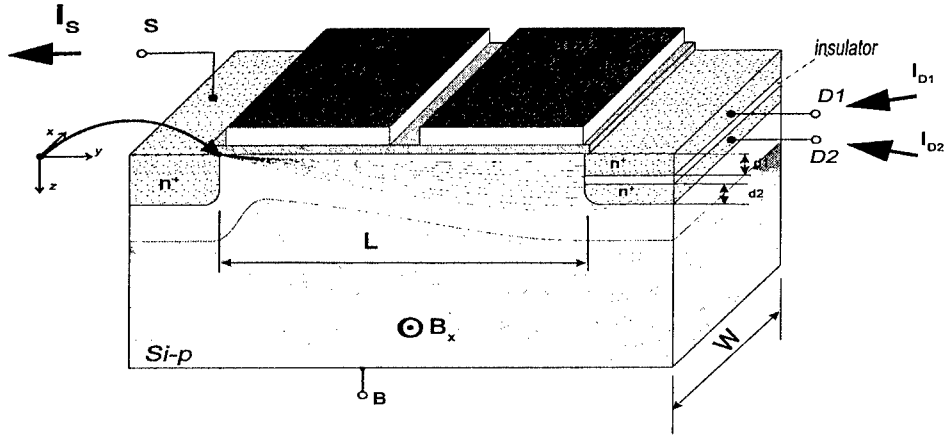
jądrowym (NMR). Równie ważnym parametrem jest moc (prąd) absorbowana przez czujnik pola magnetycznego, zwłaszcza prąd przepływający przez czujnik powinien być minimalizowany, ze względu na zakłócenia wprowadzane przez element pomiarowy. W prezentowanym artykule przedstawia się próbę rozwiązania zagadnienia konstrukcji czujnika pola magnetycznego o dużej czułości, małym poborze prądu i małych gabarytach konstrukcyjnych.

Nowa, oryginalna konstrukcja dwudrenowego tranzystora polowego z jedną lub większą liczbą bramek została zgłoszona przez Politechnikę Gdańską do ochrony patentowej w Urzędzie Patentowym RP w dniu 2001.02.22 jako projekt wynalazczy P346093 zatytułowany: *"Urządzenie do pomiaru i/lub detekcji pola magnetycznego"* [1]. Projekt ten jest bezpośrednią konsekwencją patentu RP 185178 [2]. Struktura nowego rodzaju tranzystora polowego według wynalazku przedstawiona jest szczegółowo w kilkunastu wariantach realizacji w zgłoszeniu patentowym [1]. Istota budowy tranzystora według wynalazku polega na tym, że zawiera on dwa różne obszary drenowe, wytworzone w płytce półprzewodnika i odizolowane od siebie warstwą izolatora oraz w taki sposób usytuowane jeden pod drugim w głąb półprzewodnika od jego powierzchni, że od strony obszaru źródła mają one bezpośredni kontakt z podłożem półprzewodnikowym, tworząc z nim złącza typu p-n (por. rys. 1).

W punkcie 2 zaprezentowano zasadę działania proponowanego czujnika, natomiast w punkcie 3 oszacowano jego czułość na pole magnetyczne.

2. ZASADA DZIAŁANIA NOWEGO MAGNETOTRANZYSTORA WEDŁUG WYNALAZKU [1]

Zasada działania dwudrenowego tranzystora będącego przedmiotem zgłoszenia patentowego [1] oparta jest na jednym ze zjawisk galwanomagnetycznych i jest zilustrowana na rys. 1.



Rys. 1. Dwudrenowy i dwubramkowy tranzystor MOS jako jedna z wielu wersji konstrukcyjnych czujnika pola magnetycznego według wynalazku [1].

Napięcie bramka-źródło, V_{GS} wywołuje inwersję typu przewodnictwa przy powierzchni półprzewodnika, wskutek czego zaindukowany jest kanał przypowierzchniowy, którego typ przewodnictwa jest zgodny z typem przewodnictwa źródła (S) i dwóch drenów (D_1, D_2) - na rys. 1 jest to kanał typu n. Aby zachodził przepływ elektronów od źródła tranzystora do jego drenów (D_1 i D_2) konieczna jest składowa podłużna pola elektrycznego w kanale, która jest rezultatem włączenia źródeł napięcia V_{DS1} i V_{DS2}.

Przepływ nośników prądu w kanale tranzystora polowego ma charakter dwuwymiarowy. Ten dwuwymiarowy obraz zjawisk kinetycznych w kanale tranzystora polowego został numerycznie udokumentowany, m.in. w pracach [3], [4] i [10] oraz analitycznie w pracy [5]. Na podstawie wyników tych prac można stwierdzić, że kanał tranzystora (strumień nośników prądu) odrywa się od powierzchni półprzewodnika w przydrenowej części tranzystora. Zjawisko łagodnego odrywania się kanału intensyfikuje się, gdy wartość napięcia dren-źródło wzrasta. W ten sposób całkowity prąd kanału, równy prądowi źródła I_S, rozdziela się na dwa prądy I_{D1} i I_{D2}. Możemy zatem napisać:

$$I_S = I_{D1} + I_{D2} \tag{2.1}$$

O wartości prądu źródła, I_S, decyduje wartość napięcia bramka-źródło V_{GS1}, polaryzującego bramkę $G1$, albowiem wysokość bariery potencjału dla wstrzykiwanych ze źródła do kanału elektronów modulowana jest wartością napięcia V_{GS1}. Z kolei stopień podziału prądu źródła I_S pomiędzy prądy drenu I_{D1} i I_{D2}, czyli I_{D1}/I_{D2}, zależy od wymiarów konstrukcyjnych tranzystora i stosunku napięć V_{DS1}/V_{DS2}. Na wartość stopnia podziału prądów I_{D1}/I_{D2} można wpływać bardzo efektywnie poprzez zmianę napięcia polaryzującego bramkę $G2$, V_{GS2}, dzięki czemu można równoważyć (kalibrować) czujnik w trakcie pomiarów.

Jeśli dwudrenowy tranzystor polowy umieścimy w zewnętrznym polu magnetycznym, reprezentowanym na rys. 1 przez wektor indukcji magnetycznej $\boldsymbol{B}_x$, to na poruszające się w kierunku drenów nośniki prądu (w tym przypadku elektrony) oprócz pola elektrycznego oddziaływać będzie pole magnetyczne. Ogólnie, efekt łącznego działania pola elektrycznego i magnetycznego na poruszające się nośniki ładunku zaliczamy do grupy zjawisk galwanomagnetycznych. Działania tych pól na pojedynczy ładunek elementarny opisuje wzór na siłę podany przez Lorentza:

$$\boldsymbol{F} = q\boldsymbol{E} + q(\boldsymbol{v} \times \boldsymbol{B}) \tag{2.2}$$

gdzie: q – wartość ładunku elementarnego, $\boldsymbol{E}$ – wektor natężenia pola elektrycznego, $\boldsymbol{v}$ – wektor prędkości ładunku, $\boldsymbol{B}$ – wektor indukcji pola magnetycznego.

Zatem, w wyniku włączenia bądź zmiany pola magnetycznego $\boldsymbol{B}$ trajektorie poruszających się w kanale elektronów ulegną odchyleniu "w górę" lub "w dół", co w rezultacie doprowadzi do zmiany stopnia podziału prądu źródła, I_{D1}/I_{D2}. Możemy więc napisać:

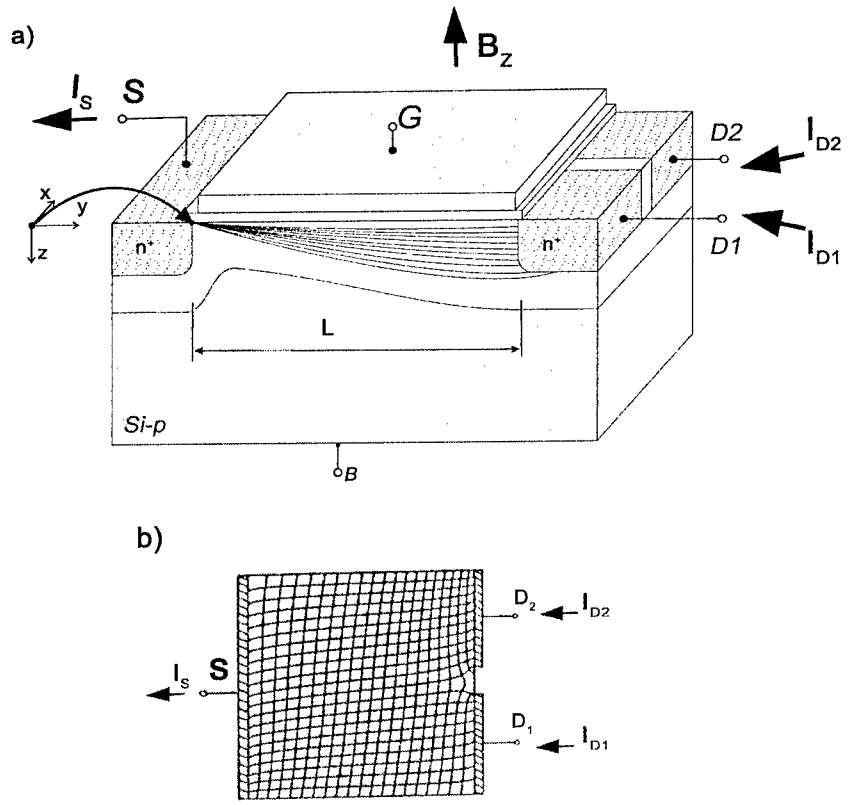
$$I_{D1} \;/\; I_{D2} = f(V_{GS}, V_{DS1}, V_{DS2}; \boldsymbol{B}) \tag{2.3}$$

oraz

$$I_{D1} \;-\; I_{D2} = k(V_{GS}, V_{DS1}, V_{DS2}; \boldsymbol{B}) . \tag{2.4}$$

Godnym uwagi jest fakt, iż w tranzystorze polowym, zwłaszcza typu MOS, prąd kanału płynie w postaci **warstw prądu** [3-5]. Największą grubość tych warstw obserwujemy przy drenach, a stosunek szerokości tych warstw (równy szerokości kanału, *W*) do ich grubości (kierunek y) jest zwykle bardzo dużą liczbą. Jak będzie dalej wykazane, właśnie ta właściwość obrazu kinetycznego nośników prądu w kanale tranzystora ma fundamentalne znaczenie dla osiągania bardzo dużych czułości czujnika (przetwornika) zbudowanego na bazie dwudrenowego tranzystora MOS. Możliwe są inne warianty konstrukcyjne dwudrenowego tranzystora spełniające funkcję czujnika pola magnetycznego – kilkanaście ich wersji konstrukcyjnych zaprezentowano w [1].

Pod względem budowy i zasady działania, nowoopracowany czujnik [1] jest najbardziej podobny do znanego w literaturze przedmiotu czujnika Popovic'a i Baltes'a, stanowiącego tranzystor SD MAGFET [8] i [9], (ang. *split-drain magnetic-field sensitive field-effect transistor*) przedstawionego na rys. 2.



Rys. 2. Czujnik SD MAGFET [8], [9] – tranzystor MOS o rozdzielonych drenach (dwudrenowy, ale inny niż na rys. 1):
a) rzut perspektywiczny,
b) kształt ugiętych polem magnetycznym linii prądu w płaszczyźnie *x-y*.

Ów tranzystor ma również dwa dreny, lecz usytuowane inaczej niż w tranzystorze według wynalazku [1]. Tranzystor SD MAGFET jest czujnikiem składowej pola magnetycznego prostopadłej do powierzchni bramki B_Z. Na skutek niezerowej wartości składowej

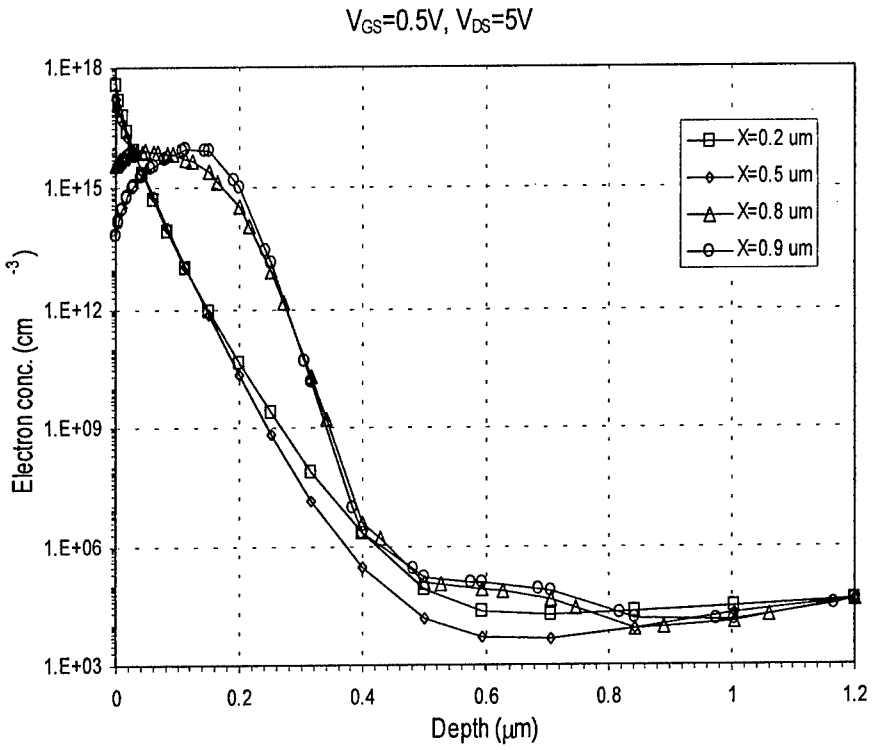
B_Z pola magnetycznego elektrony w kanale tego tranzystora odchylają się w płaszczyźnie x-y, w wyniku czego – tak jak w tranzystorze z rys. 1 – zmienia się stopień podziału prądu I_{D1}/I_{D2} (zmienia się różnica I_{D1}-I_{D2}), ale zmiana wartości stosunku I_{D1}/I_{D2} jest dużo mniejsza w przypadku tranzystora z rys. 2 niż tranzystora z rys. 1. Zaprezentowane w pracy [8] wyniki doświadczalne dotyczące przetwornika komplementarnego CMOS pozwalają oszacować pośrednio, ale z mocą quasi-eksperymentu, czułość nowego czujnika [1], co jest przedmiotem następnego rozdziału.

3. CZUŁOŚĆ NOWEGO MAGFETA

Aby wiarygodność oszacowania czułości przetwornika zbudowanego na bazie czujnika [1] była możliwie jak największa przyjęto następujące założenia, zgodne z danymi konstrukcyjnymi przetwornika zaprezentowanego w pracy [8]:

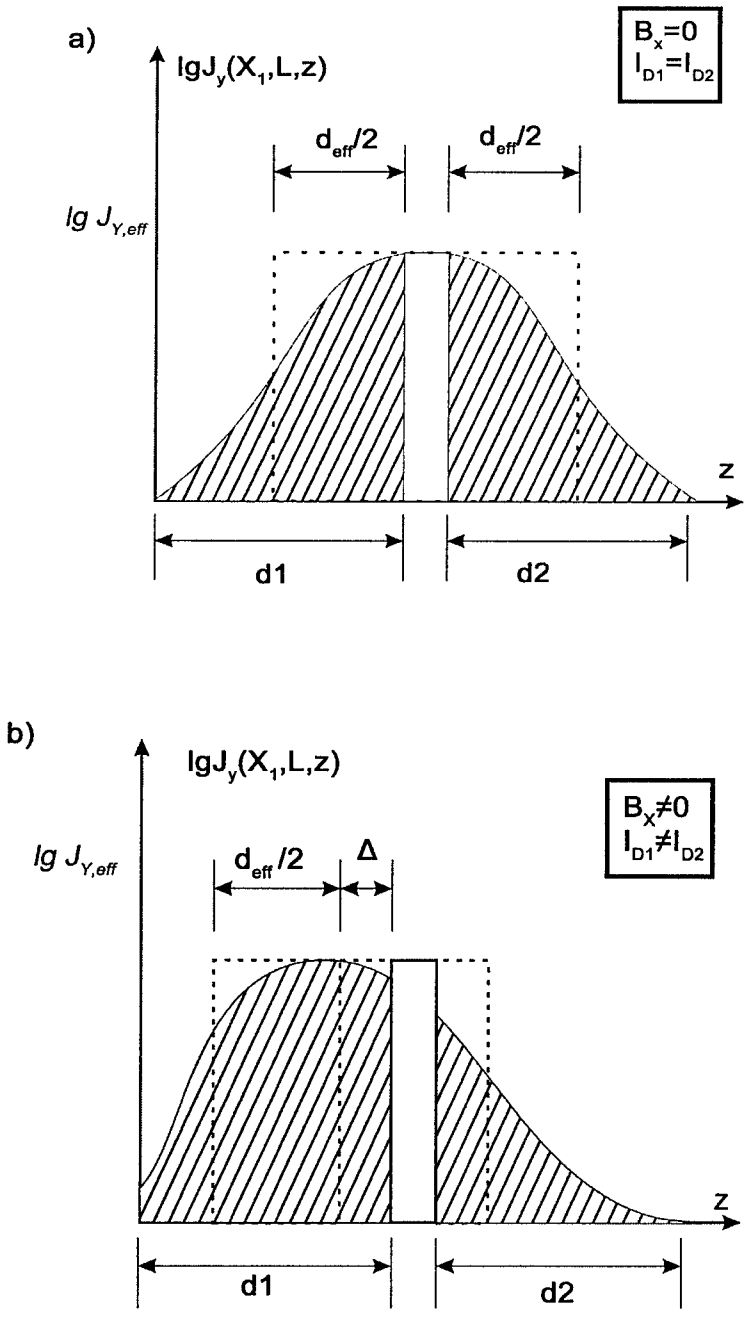
1) przetwornik według wynalazku pracuje w konfiguracji identycznej jak w pracy [8] (*Fig. 1*) – konfiguracja CMOS: w obydwu przypadkach płyną te same prądy i działają te same napięcia polaryzujące;
2) stosunek szerokości W do długości L kanału (W/L) wynosi: W/L = 22μm/14μm [8]; dla ustalenia uwagi skoncentrujemy nasze rozważania na tranzystorze z kanałem typu n;
3) odchylenie nośników prądu w kanale tranzystora wywołane działaniem odpowiedniej składowej pola magnetycznego (efekt galwanomagnetyczny) jest zjawiskiem izotropowym w obydwu konstrukcjach przetworników, to znaczy efektywne przesunięcie wiązek elektronów w obszarach przydrenowych wywołane tą samą zmianą wartości pola magnetycznego $\Delta \boldsymbol{B}$ jest w obydwu porównywanych przetwornikach identyczne (por. rys. 4 i 5);
4) efektywna średnia grubość warstwy prądu liczona w głąb półprzewodnika od jego powierzchni (rys. 3 i 4, a także rys. 5) jest porównywalna i dla obydwu przetworników wynosi $d_{eff} = 0{,}2\mu m$ (por. rys. 5)

Pewnego komentarza wymaga założenie 4). Otóż w proponowanym przetworniku (rys. 1) grubość warstwy dielektryka rozdzielającego dwa obszary drenowe powinna być rzędu kilkudziesięciu nanometrów (metoda wykonania cienkiej warstwy dielektrycznej zaproponowana będzie w odrębnym opracowaniu). Warstwa dielektryczna o tej grubości nie podważa w istotny sposób wyników symulacji otrzymanych za pomocą ATLAS'a [10] i pokazanych na rys. 3. Rys. 3 przedstawia w skali logarytmiczno-liniowej profile gęstości elektronów, a tym samym profile gęstości prądu kanału, w przekrojach prostopadłych do powierzchni półprzewodnika (w kierunku osi odciętych na rysunkach 3, 4 i 5). Widzimy, że dla przekrojów położonych bliżej obszarów drenowych prąd kanału tranzystora MOS odrywa się od powierzchni półprzewodnika, co więcej, stopień oderwania się kanału możemy ustalać za pomocą napięcia bramki G_2 (rys. 1). Parametry elektrofizyczne „symulowanego" tranzystora są następujące: grubość tlenku bramkowego = *30nm*, koncentracja domieszek w podłożu = $5E^{15}cm^{-3}$, głębokość złączy źródła/drenu = 0,2 μm, napięcie dren-źródło = *5V*, efektywne napięcie bramka-źródło = *0,5V* [10]. Na podstawie rysunku 3 możemy przyjąć, że $d_{eff} = 0{,}2\ \mu m$.

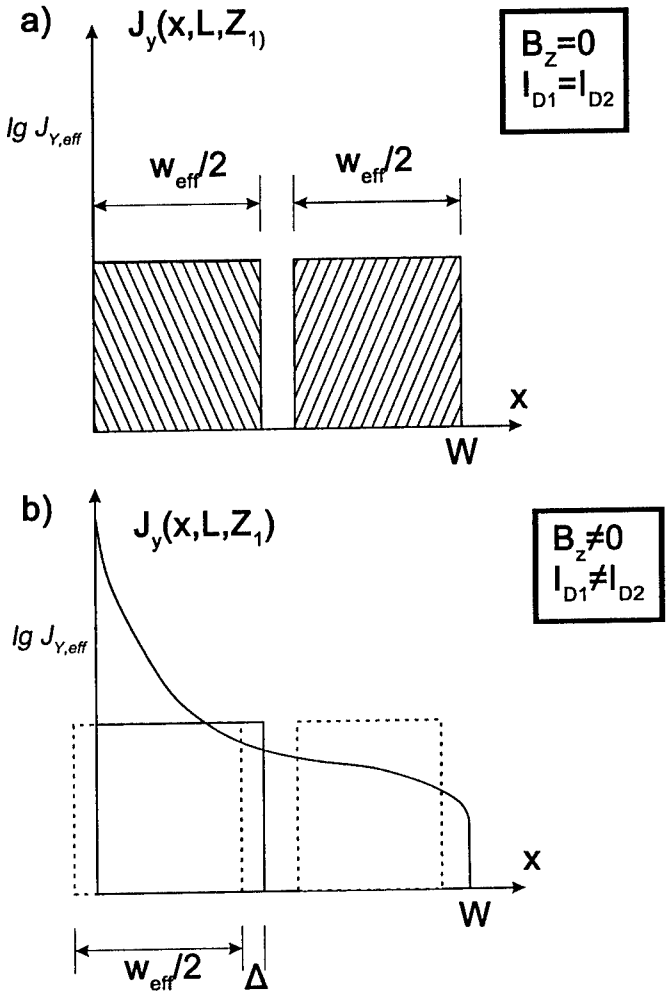


Rys. 3. Rozkłady (profile) koncentracji elektronów w typowym tranzystorze MOS w różnych przekrojach prostopadłych do kanału, $V_{DS} = 5V$, efektywne napięcie bramki $V_{GS} - V_T = 0{,}5V$; skala logarytmiczno-liniowa, x – odległość [μm] źródła tranzystora od płaszczyzny przekroju poprzecznego przez kanał, oś odciętych – odległość od tzw. międzypowierzchni tlenek/krzem, oś rzędnych – koncentracja elektronów w skali logarytmicznej.

Na rys. 4 zilustrowano w skali logarytmiczno-liniowej profile gęstości prądu kanału w obszarze przydrenowym dla nowoopracownego przetwornika [1], w przypadku: *(a)* braku pola, $B_X = 0$, oraz *(b)* w obecności pola, $B_X \neq 0$. Obecność pola magnetycznego B_X wywoła w przetworniku z rys. 1 przemieszczenie się profilu gęstości prądu o wielkość Δ – por. rys. 4. Analogiczny efekt działania pola magnetycznego na przetwornik z rys. 2 możemy zaobserwować na rys. 5 (por. rys. 2b i m.in. [9]): pojawienie się składowej pola B_Z (o wartości równej B_X w poprzednim przypadku) również wywoła efektywne przesunięcie profilu gęstości prądu o wartość Δ. Jednak skutek wywołany tym samym przesunięciem profilu gęstości prądu o wartość Δ będzie zdecydowanie różny w obydwu przetwornikach, albowiem profil z rys. 4 jest „wąski" i „wysoki" (skala logarytmiczna na osi rzędnych), a profil z rys. 5 – odwrotnie, „szeroki" i „niski" . Oznacza to, iż zmiana prądu wywołana tym samym polem magnetycznym jest dla przetwornika z rys. 1 większa niż dla przewtornika z rys. 2 (prąd jest proporcjonalny do powierzchni pola pod krzywą w układzie: gęstość prądu- oś z lub oś x). Zatem czułość przetwornika według wynalazku [1] jest wielokrotnie większa niż przetwornika przedstawionego w pracy [8].



Rys. 4. Rozkład gęstości prądu w przekroju prostopadłym do kanału tranzystora (kierunek osi z) w przetworniku z rys. 1; skala logarytmiczno-liniowa.



Rys. 5. Rozkład gęstości pradu w przekroju prostopadłym do kanału tranzystora (kierunek osi x) w przetworniku z rys.4.

Pomijając zrozumiałe rachunki i przekształcenia, można wykazać, iż czułość bezwzględna nowego przetwornika, oznaczonego symbolem *S[1]*, jest tyle razy większa od czułości przetwornika zaprezentowanego w pracy [8], *S[8]*, ile razy szerokość kanału *W* jest większa od grubości d_{eff}, co można zapisać:

$$S[1] = (W / d_{eff})\, S[8] \tag{3.1}$$

Ta sama reguła odnosi się do czułośći względnej (ang. *relative sensitivity*, S_R):

$$S_R[1] = (W / d_{eff})\, S_R[8] \tag{3.2}$$

Wykorzystując wyniki pomiarowe i dane konstrukcyjne przetwornika z pracy [8] możemy przewidzieć czułość nowego przetwornika $S[1]$ i $S_R[1]$:

$$S[1] = (22/0,2)\, S[8] = 110\, S[8] \approx 130\ [V/T] \tag{3.3}$$

$$S_R[1] > 10^6\ [V/(A \cdot T)] \tag{3.4}$$

Zaproponowany przetwornik pola magnetycznego ma tę przewagę nad przetwornikiem z pracy [8], że jego czułość bezwzględną można wydatnie zwiększyć poprzez zwiększenie szerokości kanału W tranzystora, czego nie można osiągnąć w przetworniku z pracy [8]. Wydaje się więc możliwe wykonanie przetwornika o czułości ponad 1000-krotnie większej (ponad 1000 V/T) niż czułość przetwornika zaprezentowanego w pracy [8].

4. ZAKOŃCZENIE

Przedstawione wyniki analizy porównawczej dwóch podobnych typów przetworników opartych na różnych konstrukcjach dwudrenowych tranzystorów MOS (por. rys. 1 i 2), wskazują jednoznacznie na celowość podjęcia badań empirycznych nad nowym przetwornikiem pola magnetycznego, zgłoszonym do ochrony patentowej. Przewidywana czułość bezwzględna nowego przetwornika jest obiecująca, a jej wartość może być większa od 1000 [V/T].

BIBLIOGRAFIA

[1] Kordalski W., Polowczyk M., Czerwiński J., Dzierżko J. Dobrzański L.: *Urządzenie do pomiaru i/lub detekcji pola magnetycznego*, Zgłoszenie patentowe P346093 z dnia 22.02.2001r.

[2] Kordalski W.: *Dwubramkowy tranzystor polowy typu MOS z kanałem indukowanym*, patent RP nr.185178 (30.10.2002).

[3] Yamaguchi K.: *Field-Dependent Mobility Model for Two-Dimensional Numerical Analysis of MSFET's*, IEEE Trans. on Electron Devices, Vol. ED-26, No 7, 1979.

[4] Selberherr S.: *Analysis and Simulation of Semiconductor Devices*, Wien, Springer-Verlag, 1984.

[5] Kordalski W.: *Two-Dimensional Effects in the MOSFET*, Proc. of the XIX-th National Conference on Circuit Theory and Electronic Networks, Kraków-Krynica, Poland, 1996, Vol. II, pp. 239-244.

[6] Kobus A., Tuszyński J., Warsza Z.L.: *Technika hallotronowa*, WNT, Warszawa, 1980.

[7] Kirijew P. S.: *Fizyka półprzewodników*, WNT, Warszawa, 1971.

[8] Popović R.S., Baltes H.P.: *A CMOS Magnetic Field Sensor*, IEEE J. of Solid-State Circuits, Vol. SC-18, No.4, 1983.

[9] Baltes H.P., Popović R.S.: *Integrated Semiconductor Magnetic Field Sensor*, Proceedings of the IEEE, Vol. 74, No 8, 1986 pp.1107-1132.

[10] Tomaszewski D., Instytut Technologii Elektronowej, Warszawa, *Symulacje za pomocą ATLASA* – informacja prywatna.

A NEW TWO-DRAIN MOS TRANSISTOR AS A MAGNETIC FIELD SENSOR

Summary

A novel, two-drain MOS transistor as a very sensitive magnetic field sensor is proposed. The sensor exploits the galvanomagnetic effect due to the Lorentz force on charge carriers in the transistor channel.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Wiesław Kordalski [*], Tomasz Stefański [#]

Katedra Systemów Mikroelektronicznych, Politechnika Gdańska
[#] Centrum Techniki Morskiej, Gdynia

ADMITANCJA WEJŚCIOWA I TRANSADMITANCJA MOSFETA W ZAKRESIE B.W.CZ.

Streszczenie

Bazując na nowym, nie quasi-statycznym modelu małosygnałowym tranzystora MOS, przeanalizowano i zweryfikowano eksperymentalnie aż do *30 GHz* małosygnałową admitancję wejściową i transadmitancję tego elementu - dwa ważne i budzące najwięcej wątpliwości parametry macierzy admitancyjnej tranzystora.

1. WSTĘP

Rozwój technik i urządzeń do przesyłania informacji w zakresie coraz to wyższych częstotliwości intensyfikuje zainteresowania projektantów scalonych układów MOS dokładnymi i adekwatnymi modelami tranzystorów polowych dla zakresu b.w.cz. Znane w literaturze przedmiotu modele małosygnałowe **tranzystora wewnętrznego** wyprowadzone są na bazie zbyt daleko idących uproszczeń. Na przykład, nie są znane modele, w których jednocześnie uwzględniono by zależność ruchliwości nośników prądu w kanale tranzystora od wartości pola elektrycznego i elektryczne sprzężenie pomiędzy kanałem a strukturą (bramką i podłożem). Często w projektowaniu układów scalonych pracujących w zakresie częstotliwości radiowych i mikrofalowych stosuje się tak zwane modele quasi-statyczne, które dają w rezultacie *zerowy moduł admitancji wejściowej tranzystora wewnętrznego*, niezależnie od częstotliwości, ale za to *moduł transadmitancji rośnie do nieskończoności*, gdy częstotliwość sygnału wzrasta (por. model Enz'a i Cheng'a [1], 2000 r., a także przedstawiony w pracy [2]) – co jest wynikiem nierealistycznym. Tymczasem, wraz ze wzrostem częstotliwości sygnałów elektrycznych przenoszonych przez tranzystor MOS coraz większą rolę odgrywają, nie uwzględniane we wcześniejszych analizach tego tranzystora, zjawiska inercyjne w transporcie nośników prądu w kanale tranzystora. Jak wiadomo, nośniki ładunku w półprzewodnikach poruszają się jedynie ze skończoną prędkością, na przykład prędkość nasycenia (maksymalna) elektronów w krzemie wynosi około 10^7 *cm/s*. Nie można więc przyjmować – tak, jak to robiono dotychczas w modelach quasi-statycznych – że prędkość ta jest nieskończona, ponieważ nie uwzględ-

dnia się wtedy opóźnienia prądu drenu względem prądu źródła w kanale tranzystora MOS. Z faktu tego wynika, że czas przelotu, zdefiniowany zależnością:

$$\tau = \int_0^L \frac{dy}{v(y)}$$

w której L jest długością kanału, $v(y)$ prędkością nośników wzdłuż kanału, ma znaczenie podstawowego kryterium ilościowego w dziedzinie częstotliwościowej analizy pracy tranzystora. Dlatego dla τ większego od około 1/10 okresu sygnału harmonicznego pobudzającego tranzystor niezbędna jest analiza nie quasi–statyczna (ang. NQS: Non-Quasi-Static), natomiast gdy τ jest krótszy od wspomnianej wartości, to może być stosowana analiza quasi–statyczna. U podstaw małosygnałowej analizy quasi–statycznej leży założenie, że reakcja tranzystora MOS, rozumiana jako zmiana prądu drenu na przyłożone napięcie bramka–źródło, jest natychmiastowa, bezzwłoczna. Innymi słowy, wszelkie procesy ustalania się równowagi dynamicznej zachodzą w czasie wielokrotnie krótszym od okresu wymuszającego sygnału harmonicznego. Z kolei w analizie czy modelowaniu nie quasi–statycznym uwzględnia się skończoną prędkość nośników prądu i skończony czas potrzebny na rekonfigurację stanu elektrofizycznego tranzystora. Na przykład, analizując pracę układu, w którym występuje chociaż jeden tranzystor MOS o bardzo długim kanale, ok. *50 μm*, należy zastosować model NQS już w zakresie częstotliwości rzędu *1 MHz*.

Znane z literatury przedmiotu nie quasi–statyczne modele małosygnałowe tranzystora wewnętrznego możemy podzielić na trzy klasy:

1. Modele, w których moduł transadmitancji y_m jest funkcją stałą częstotliwości (zdecydowana większość modeli *książkowych* plus model z pracy [3,4]):

$$|y_m(j\omega)| = const$$
$$\lim_{\omega\to\infty} |y_m(j\omega)| = const$$

2. Modele, w których moduł transadmitancji jest funkcją rosnącą (!) częstotliwości [1,2]:

$$\lim_{\omega\to\infty} |y_m(j\omega)| = \infty$$

3. Modele w postaci linii *RC* o stałych rozłożonych, dla których spełniony jest warunek (m. in. BSIM3 [5]):

$$\lim_{\omega\to\infty} |y_{21}(j\omega)| = 0$$

ale **nie spełniony** jest warunek unilateralności, gdyż dla tych modeli

$$|y_{12}(j\omega)| \neq 0$$

Kompletny model tranzystora, model tranzystora rzeczywistego, musi być uzupełniony o elementy pasożytnicze: rezystory R, pojemności C i indukcyjności L, których wartości i wpływ na charakterystyki częstotliwościowe tranzystora zależy od parametrów procesu technologicznego i konstrukcji tranzystora.

Modele z grupy 1 i 2 spełniają wprawdzie warunek unilateralności, oznaczający że transport mocy sygnału odbywa się z obwodu bramka–źródło do obwodu dren–źródło, ale nie spełniają oczywistego warunku asymptotycznego polegającego na zaniku zdolności przenoszenia energii z obwodu wejściowego do wyjściowego tranzystora w miarę wzrostu częstotliwości, co jest równoważne warunkowi:

$$\lim_{\omega\to\infty}\left|y_m(j\omega)\right| = 0$$

Modele grupy 3, odwrotnie, spełniają warunek asymptotyczny dla $|\ y_m(j\omega)\ |$, ale nie spełniają warunku unilateralności. W podsumowaniu można stwierdzić, że każdy ze znanych modeli nie spełnia jednocześnie dwóch warunków koniecznych spójności fizycznej modelu małosygnałowego tranzystora MOS: (i) warunku unilateralnosci, i (ii) konieczności malenia do zera modułu transadmitancji w funkcji częstotliwości, co w konsekwencji prowadzi do bardzo dużych błędów przy próbie opisu teoretycznego pomierzonych charakterystyk częstotliwościowych elementów macierzy admitancyjnej *Yij* za pomocą znanych modeli.

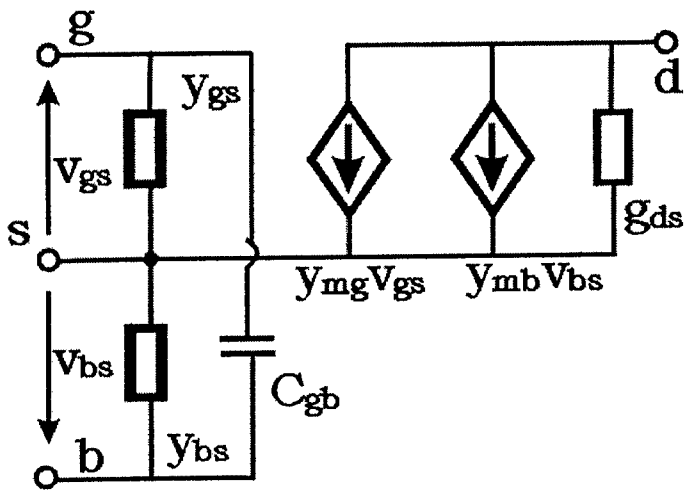
Celem proponowanego referatu jest zaprezentowanie nowego, nie quasi-statycznego modelu małosygnałowego tranzystora MOS. Nowy model spełnia wszystkie warunki konieczne spójności fizycznej, obowiązuje od *0 Hz* do częstotliwości mikrofalowych włącznie oraz daje realistyczne wyniki ilościowe. W odróżnieniu od modelu BSIM3v3, daje bardzo dobrą zgodność wyników teoretycznych z eksperymentalnymi aż do ok. *30 GHz*. W referacie szczególny akcent położono na badanie admitancji wejściowej i transadmitancji tranzystora w funkcji częstotliwości.

2. NIE QUASI-STATYCZNY MODEL MAŁOSYGNAŁOWY TRANZYSTORA WEWNĘTRZNEGO [6]

Fizyczne i matematyczne podstawy niezbędne do opracowania nowego, nie quasi-statycznego modelu małosygnałowego czterokońcówkowego tranzystora wewnętrznego MOS sformułowane zostały w pracy [6]. Przedstawiono tam układ równań różniczkowych cząstkowych definiujących nowy model NQS w dziedzinie czasu. Spójność fizyczną modelu osiągnięto dzięki rozważeniu tranzystora wewnętrznego jako obiektu dwuwymiarowego, do którego z obszaru źródła wstrzykiwane są nośniki prądu traktowane jako ciągłe ściśliwe medium ładunku (ściśliwa ciecz ładunku). W odróżnieniu od innych modeli, ruchliwość wstrzykiwanych do kanału nośników jest zależna od wartości składowej podłużnej pola elektrycznego unoszącej nośniki pomiędzy źródłem a drenem tranzystora, w rezultacie czego efekt nasycenia prędkości nośników prądu jest z założenia uwzględniony. Ponadto założono, że wstrzyknięte do kanału nośniki nadmiarowe oddziaływają elektrycznie z bramką i podłożem, indukując w tych obszarach ładunki elektryczne – ma miejsce sprzężenie elektryczne pomiędzy zaburzoną koncentracją nośników prądu kanału a strukturą tranzystora (bramką i podłożem). W wyniku tego sprzężenia w bramce i podłożu tranzystora indukują się stosowne prądy, które w dotychczasowych teoriach i modelach tranzystora MOS nie były uwzględniane. To właśnie dzięki temu sprzężeniu można mówić o różnej od zera admitancji wejściowej tranzystora wewnętrznego, spełniając przy tym jednocześnie warunek unilateralności tranzystora. Takiego rezultatu nie można osiągnąć posługując się metodologią zastosowaną, m. in., do wyprowadzenia modelu BSIM3v3 [5] czy modelu przedstawionego w pracy [4]. Autorzy tych prac zakładają, że oddziaływanie warstwy prądu kanału z bramką czy podłożem zachodzi według mechanizmu właściwego liniom *RC* o stałych rozłożonych, który to mechanizm i tak został

zredukowany przez autorów tych prac do zastąpienia linii *RC* dwójnikami *RC*. Jak wiadomo, zarówno linia *RC*, jak i dwójnik *RC* nie są elementami unilateralnymi. Jak wykazano w pracy [6], transport nośników wstrzykniętych przez barierę potencjału do kanału zachodzi według dwóch mechanizmów: dyfuzyjnego i unoszenia (dryfu) w stałym polu elektrycznym wynikającym polaryzacji napięciowej tranzystora. Także w pracy [6] wykazano, że wyrażenie na całkowitą gęstość prądu składowej zmiennej kanału tranzystora zawiera oprócz składników opisujących dwa wyżej wspomniane mechanizmy jeszcze jeden, który jest rezultatem istnienia małosygnałowej składowej podłużnej pola elektrycznego w kanale.

Na podstawie równań zaprezentowanych w pracy [6] opracowano małosygnałowy, nie quasi-statyczny model i schemat zastępczy tranzystora MOS w dziedzinie częstotliwości. Wyniki tego opracowania przedstawiono w pracy [7]. Na rys. 1 pokazano równoważny schemat zastępczy **wewnętrznego tranzystora MOS**, odpowiadający analizie nie quasi-statycznej.



Rys. 1. Tranzystor wewnętrzny MOS – elektryczny schemat zastępczy.

Elementy schematu zastępczego zdefiniowane są następującymi formułami:

$$\mu(E_0)=\frac{\mu_0}{[1+(E_0/E_C)^{\beta}]^{1/\beta}},\left(E_0=\frac{|V_{DS}|}{L}\right), \tag{2.1}$$

$$C_{gb}=\frac{g_m\eta L}{(1+\eta)\mu(E_0)E_0},\left(\eta=\frac{g_{mb}}{g_m}\right), \tag{2.2}$$

$$Y_{mg}=g_m\exp(\gamma L),\qquad Y_{mb}=g_{mb}\exp(\gamma L), \tag{2.3}$$

$$Y_{gs}=j\omega\frac{C_{gb}D_C[\exp(\gamma L)-1]}{\eta\gamma L(1-\gamma V_T/E_0)}, \tag{2.4}$$

$$Y_{bs} = \eta^2 Y_{gs}\,, \tag{2.5}$$

$$\gamma = \frac{1}{2}\left(\frac{E_0}{V_T} - \sqrt{\frac{\sqrt{a^2+b^2}+a}{2}} - j\sqrt{\frac{\sqrt{a^2+b^2}-a}{2}} \right), \tag{2.6}$$

$$a = \left(\frac{E_0}{V_T} \right)^2, b = \frac{4\omega (1 + D_C)}{V_T \mu (E_0)}, V_T = \frac{kT}{q}, \tag{2.7}$$

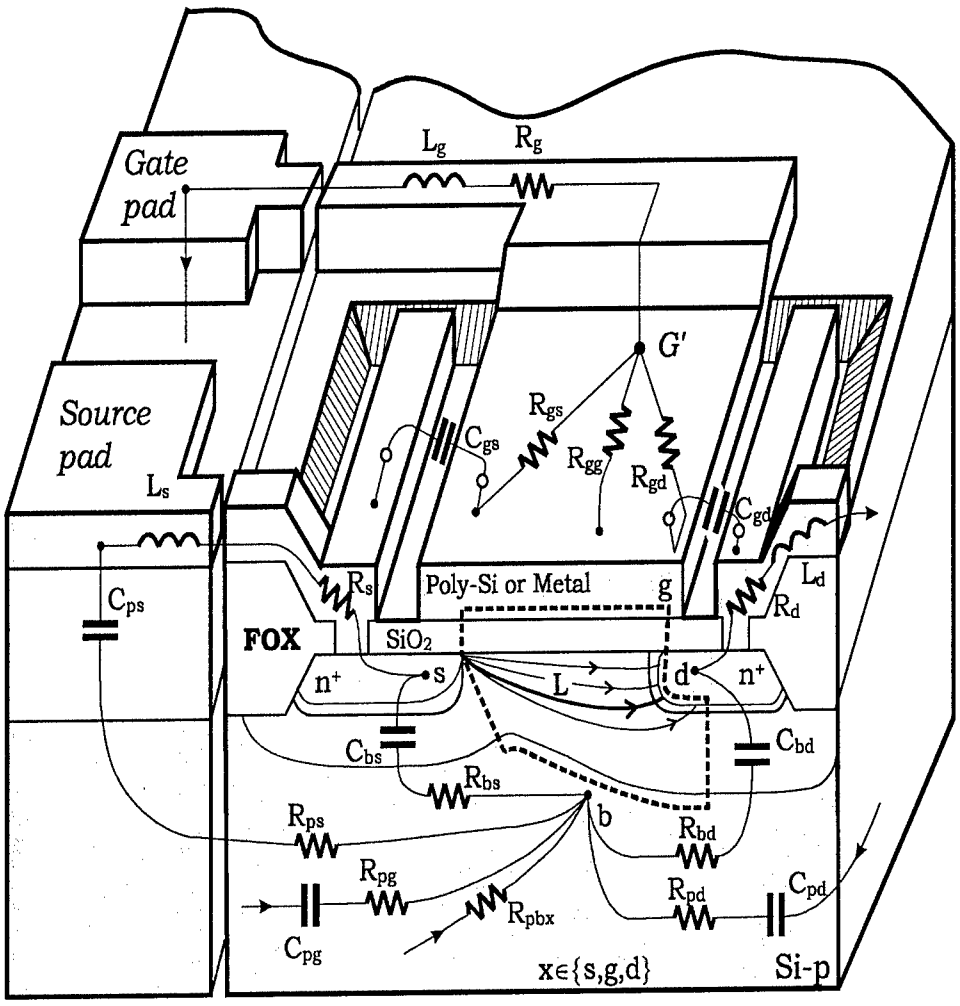
w których g_{ds} , g_m i g_{mb} są , odpowiednio, niskoczęstotliwościową (quasi-statyczną) konduktancją dren-źródło, transkonduktancją bramki i transkonduktancją podłoża w punkcie pracy tranzystora. μ_o , E_C i E_0 oznaczają, odpowiednio, ruchliwość niskopolową, wartość pola charakterystycznego i wartość pola podłużnego w kanale tranzystora w jego punkcie pracy. D_C jest dynamicznym współczynnikiem sprzężenia pomiędzy kanałem a bramką i podłożem (D_C nie większy od 1), L – długością kanału, a V_{DS} napięciem dren-źródło polaryzującym tranzystor.

Z równań (2.3) – (2.7) wynika, że część rzeczywista współczynnika propagacji γ jest malejącą funkcją częstotliwości ω, co implikuje malenie transadmitancji y_{mg} i y_{mb} ze wzrostem częstotliwości sygnału sterującego bramką. Nowymi elementami, nie występującymi w schematach zastępczych znanych modeli małosygnałowych, są admitancje Y_{gs} i Y_{bs}, które reprezentują elektryczne sprzężenie pomiędzy kanałem a strukturą tranzystora.

3. KOMPLETNY SCHEMAT ZASTĘPCZY TRANZYSTORA I JEGO WERYFIKACJA EKPERYMENTALNA

Małosygnałowy schemat zastępczy rzeczywistego tranzystora MOS, zwłaszcza dla zakresu częstotliwości radiowych i mikrofalowych, musi zawierać w sobie elementy pasożytnicze, które łączą tranzystor wewnętrzny z dostępnymi zaciskami zewnętrznymi rzeczywistego tranzystora. Elementy pasożytnicze, sposób ich reprezentacji jest problemem samym sobie złożonym, szczególnie gdy analizujemy pracę tranzystora w obszarze częstotliwości kilkudziesięciu *GHz*. Złożoność układu reprezentującego elementy pasożytnicze zależy od konkretnej sytuacji badawczej. Można na przykład sieć elementów pasożytniczych znacznie zredukować, jeśli zaprojektuje się odpowiednio topografię tranzystora badanego – stosownie do dostępnej aparatury pomiarowej i odpowiednich technik deembeddingu. Spośród zagadnień dotyczących elementów pasożytniczych występujących w tranzystorze MOS, wydaje się, że problem adekwatnego modelowania rezystancji i pojemności rozproszonej bramki jest dużym wyzwaniem dla badaczy, zwłaszcza w tych przypadkach, w których długość kanału tranzystora jest rzędu *100 nm*.

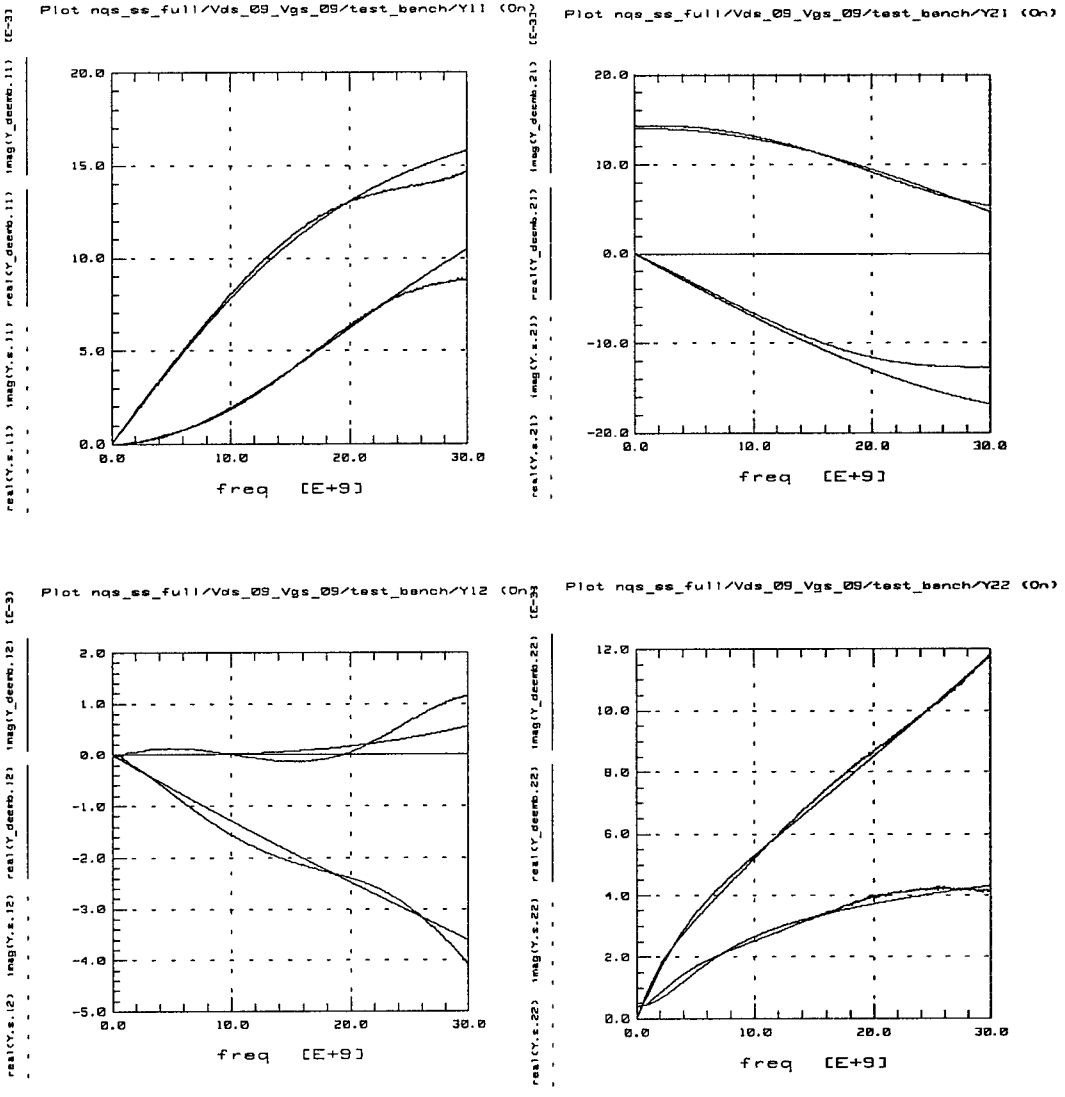
Na rys. 2 przedstawiono bryłę tranzystora MOS, w której wyróżniono tranzystor wewnętrzny i możliwie kompletną sieć elementów pasożytniczych.



Rys. 2. Widok ogólny tranzystora MOS z wyróżnionym tranzystorem wewnętrznym i elementami pasożytniczymi.

Nowy, nie quasi-statyczny model będący przedmiotem niniejszej pracy został już pozytywnie zweryfikowany eksperymentalnie na wielu zbiorach danych doświadczalnych. W naszym artykule [7] zaprezentowaliśmy bardzo dobrą zgodność tego modelu z danymi empirycznymi dostępnymi w literaturze przedmiotu [4] (kompletne charakterystyki częstotliwościowe wszystkich elementów macierzy admitancyjnej aż do *27 GHz* tranzystora o długości kanału $L = 0.4\ \mu m$). Zgodność wyników doświadczalnych z wynikami teoretycznymi uzyskanymi zarówno według modelu autorskiego [4], jak i na podstawie modelu BSIM3v3 [4] nie zbliżają się nawet do poziomu dokładności opisu charakterystyk doświadczalnych osiągniętego za pomocą modelu zaprezentowanego w tym artykule – por. wyniki pracy [4] i [7]. Niezależnie od wspomnianej weryfikacji, w Instytucie Fraunhofera (Erlangen, Niemcy) jeden ze współautorów (T. S.) zaprojektował i dokonał serii pomiarów

w odpowiednich warunkach za pomocą najnowocześniejszej aparatury pomiarowej w zakresie częstotliwości do *30 GHz*. Wstępne wyniki ilustrujące zgodność modelu nie quasi-statycznego z danymi eksperymentalnymi tranzystora dziesięcio-palczastego ($F = 10$) uzyskanymi w Instytucie Fraunhofera zaprezentowano na rys. 3.



Rys. 3. Pomierzone i symulowane małosygnałowe charakterystyki częstotliwościowe elementów macierzy admitancyjnej tranzystora o długości kanału $L=0.35\mu m$, szerokości kanału $W=100\mu m$ i strukturze dziesięciopalczastej ($F=10$).

Na podstawie wyników z rys. 3 można stwierdzić, że zgodność opisu teoretycznego otrzymanego za pomocą modelu nie quasi-statycznego z wynikami eksperymentalnymi jest bardzo dobra.

6. ZAKOŃCZENIE

Na bazie nowego, nie quasi-statycznego modelu małosygnałowego tranzystora MOS dokonano analizy admitancji wejściowej i transadmitancji – dwu ważnych parametrów charakteryzujących pracę tranzystora w obszarze b. w. cz. Zwrócono uwagę na niedoskonałości występujące w opisie teoretycznym pracy tranzystora. Przedstawiono podstawowe równania definiujące nowy model oraz wykazano, że zaproponowany model tranzystora wewnętrznego usuwa podstawowe wady znanych w literaturze przedmiotu modeli. Przydatność nowego modelu do bardziej efektywnego projektowania układów scalonych pracujących w zakresie częstotliwości radiowych i mikrofalowych potwierdzono bardzo dobrymi wynikami weryfikacji eksperymentalnej.

BIBLIOGRAFIA

[1] Enz C. C., Cheng Y.: *MOS Transistor Modelling for RF IC Design*, IEEE Trans. on Solid-State Circuits, Vol.35, No.2, 2000, pp. 186-201.
[2] Sallese J. M., Porret A. S.: *A Novel Approach to Charge-Based Non-Quasi-Static Model of the MOS Transistor Valid in all Modes of Operation*, Solid-State Electronics, Vol.44 (2000) pp. 887-894.
[3] Manku T.: *Microwave CMOS-Device Physics and Design*, IEEE J. of Solid-State Circuits, Vol.34, No.3, 1999, pp. 277-285.
[4] Sung R. et. al.: *Extraction of High-Frequency Equivalent Circuit Parameters of Submicron Gate-Length MOSFET's*, IEEE Trans. on Electron Devices, Vol.45, No.8, 1998, pp. 1769-1775.
[5] Chan M. et. al.: *A Robust and Physical BSIM3 Non-Quasi-Static Transient and AC Small-Signal Model for Circuit Simulation*, IEEE Trans. on Electron Devices, Vol.45, No.4, 1998, pp. 834-841.
[6] Kordalski W.: *An Injection Non-Quasi-Static Small-Signal MOSFET Model*, International Conference on Signals and Electronic Systems, 17-20 October 2000, 2000, Ustroń, Poland.
[7] Kordalski W., Stefański T.: *A Non-Quasi-Static Small-Signal MOSFET Model for Radio and Microwave Frequencies Including Spreading Gate Resistances and Capacitances,* Proceedings of the 2003 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium, Philadelphia, PA, June 8-10, 2003, pp.365 – 368.

INPUT ADMITTANCE AND TRANSADMMITANCE OF THE MOSFET FOR VERY HIGH FREQUENCIES

Summary

Some results of an analysis of the input admittance and transadmittance of the MOSFET operating over very high frequency range are presented. The results are derived from a novel non-quasi-static small-signal MOSFET model, and are confirmed by experimental verification.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Slawomir Koziel

Department of Microelectronics Systems, Gdansk University of Technology

CONTINUOUS-TIME ACTIVE-RC FILTERS: GENERAL MODEL AND ITS APPLICATIONS

Summary

In the paper, a general model of integrator-based continuous-time Active-RC filter is presented. More specifically, filter structures containing inverting amplifiers and passive feedback network are considered. The structure is analyzed using matrix description. The explicit transfer function formula is given in general case. The extensions of the model that take into account finite DC gain and finite bandwidth of operational amplifiers as well as other non-ideal effects are discussed. The matrix-based approach is formulated especially for efficient use in computer-aided analysis and design of Active-RC filters. An application example of the proposed approach to obtain OPAMP design specifications for 3rd order low-pass Chebyshev filter for VDSL Analog Front End are given.

1. INTRODUCTION

Continuous-time analog filters based on operational amplifiers (OPAMPs) and RC elements (Active-RC filters) provide solutions for various signal-processing tasks. Many synthesis and design methods for different types and architectures of this class of filters have been reported [1]-[3]. Although other techniques, particularly transconductance-capacitor (OTA-C) filters [4], are now dominant in high-frequency range, Active-RC filters are suitable for numerous medium-frequency applications, especially those which require high linearity and low noise, such as ADSL [5], VDSL [6], [7], WCDMA [8], [9], RFIC receivers for PANs [10], GSM baseband transmitters [11]. Recently, an Active-RC filter for frequencies beyond 100MHz has been successfully implemented [12].

In this paper, we consider a general Active-RC filter topology suitable for computer-aided analysis, design and optimization of Active-RC filters. The structure is not the most general one, since it is restricted to filter topologies containing only inverting amplifiers (especially integrators) and RC feedback network. Thus, it is not suitable for analyzing some classical structures such as Sallen-Key biquad [3]. However, it covers a number of practical Active-RC filters, especially most multiple-loop feedback topologies including filters based on RLC ladder simulation, leap-frog, follow-the-leader [1]-[3], and

This work was supported in part by the State Scientific Research Committee, Poland, under Grant 4T11B01625

makes it possible to generate and investigate new filter topologies with possibly attractive properties.

2. INTEGRATOR-BASED ACTIVE-RC FILTERS

Fig.1 shows the single-ended version of the general topology of integrator-based Active-RC filter. Note that we consider filter topologies containing somewhat more general building blocks than lossy or lossless integrators, i.e. inverting amplifiers that may contain nonzero capacitance at the input. The filter in Fig.1 contains n operational amplifiers denoted as O_i, i=1,...,n, n input resistors R_{bi} and capacitors C_{bi}, i=1,...,n, the set of internal feedback resistors R_{ij} and capacitors C_{ij}, i,j=1,...,n, as well as output summer consisting of amplifier O_o, resistors R_d (direct feedforward path from input), R_{ci}, i=1,...,n and R_o. Output nodes of OPAMPs are denoted as x_i, i=1,...,n. We shall also use U_{zi} and U_{xi} to denote voltages at OPAMP input and output nodes. Any integrator-based Active-RC filter is a particular case of the structure in Fig.1 (after removing appropriate elements from the schematic diagram and setting corresponding capacitance [resistance] values to zero [infinity]). In this section we shall assume that operational amplifiers are ideal (we relax this assumption in Section 3, where filters with non-ideal OPAMPs are considered).

Let us start the analysis of the circuit in Fig.1. The total current flowing into each node z_i, i=1,...,n is zero and the voltage at z_i is also zero, due to the assumed ideality of OPAMP. This allows us to write the following equations:

$$y_{bi}U_{in} + \sum_{j=1}^{n} y_{ij}U_{xj} = 0, \qquad i=1,...,n \tag{2.1}$$

$$g_o U_{out} + \sum_{j=1}^{n} g_{cj}U_{xj} + g_d U_{in} = 0 \tag{2.2}$$

where U_{in} (U_{out}) denote Laplace transform of input (output) voltage, and

$$y_{bi} = g_{bi} + sC_{bi}, \quad g_{bi} = 1/R_{bi}, \quad i=1,...,n, \quad g_{ci}=1/R_{ci}, \quad i=1,...n \tag{2.3}$$

$$y_{ij} = g_{ij} + sC_{ij}, \quad g_{ij} = 1/R_{ij}, \quad i,j=1,...,n, \quad g_d=1/R_d, \quad g_o=1/R_o, \tag{2.4}$$

Let us introduce the following notation (symbol ^{T} denotes matrix transposition)

$$\begin{aligned} &\boldsymbol{T} = [C_{ij}]_{i,j=1}^{n}, \quad \boldsymbol{G} = [g_{ij}]_{i,j=1}^{n}, \quad \boldsymbol{X} = [U_{x1} \ \cdots \ U_{xn}]^T \\ &\boldsymbol{C} = [c_1 \ \cdots \ c_n] = [g_{c1}/g_o \ \cdots \ g_{cn}/g_o], \\ &\boldsymbol{B} = [g_{b1} + sC_{b1} \ \cdots \ g_{bn} + sC_{bn}]^T, \quad \boldsymbol{D} = d = g_d/g_o \end{aligned} \tag{2.5}$$

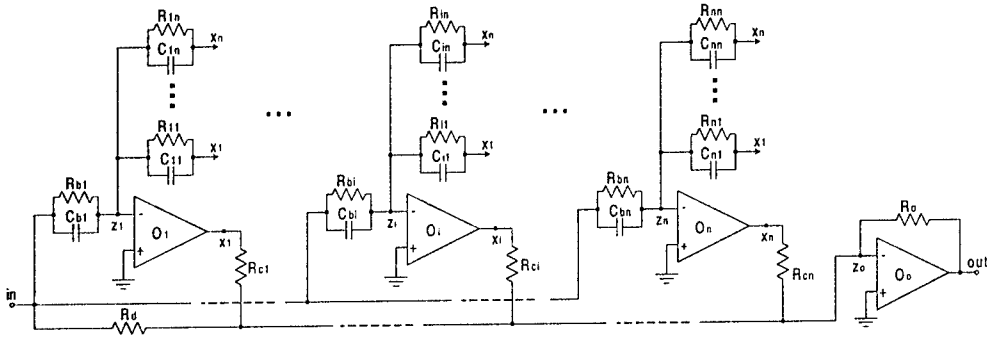


Fig.1. General structure of integrator-based Active-RC filter

Using (2.5) we can rewrite (2.1), (2.2) in the following form

$$\begin{aligned} (sT+G)X + BU_{in} &= 0 \\ U_{out} + CX + DU_{in} &= 0 \end{aligned} \tag{2.6}$$

This allows us to calculate the transfer function $H(s)$ of the filter

$$H(s) = \frac{U_{out}(s)}{U_{in}(s)} = C(sT+G)^{-1}B - D \tag{2.7}$$

Now, let us denote adjoint matrix of $sT+G$ as A where

$$A(s) = adj(sT_C + G) = [A_{ij}(s)]^T_{i,j=1} \tag{2.8}$$

This allows us to rewrite transfer function in the form:

$$H(s) = \frac{1}{\det(sT+G)} \sum_{i,j=1}^{n} c_i (g_{bj} + sC_{bj}) A_{ij}(s) - d \tag{2.9}$$

Note that in many filter structures, input signal is provided to only one internal node, say z_k (i.e. there is no input signal distribution), and there is no output summer (i.e. one of the internal nodes, say x_l, is the output of the filter), and no input capacitors. This means that $B=[0 \ \cdots \ 0 \ g_{bk} \ 0 \ \cdots \ 0]^T$, $C=[0 \ \cdots \ 0 \ -1 \ 0 \ \cdots \ 0]$ with 1 at l-th position and $C_{bi}=0$ for $i=1,...n$ (note that we have -1 because of the fact that if there is no output summer, there is no signal inversion due to O_o either). In such case, expression (2.9) reduces to the form:

$$H(s) = -\frac{g_{bk} A_{lk}(s)}{\det(sT+G)} \tag{2.10}$$

Similar expression can be written for slightly more general case, with no input capacitors, input signal distribution (i.e. $B = [g_{b1} \ \cdots \ g_{bn}]$), and a trivial output summer ($C = [0 \ \cdots \ 0 \ -1 \ 0 \ \cdots \ 0]$- -1 at l-th position). Then, we have

$$H(s) = -\frac{1}{\det(sT_C - G)} \sum_{i=1}^{n} g_{bi} A_{li}(s) \tag{2.11}$$

On the basis of the above expressions one can easily calculate the transfer function of any particular structure of integrator-based Active-RC filter. It is also worth noting that there is one-to-one correspondence between integrator-based Active-RC filters and matrices (2.5). Thus, using presented matrix description we can consider filter design, analysis and optimization in purely algebraic domain. The main advantage is that this can be easily handled by a suitable computer software.

It follows that the order n_H of transfer function of general filter structure in Fig.1 is not necessarily equal to the number of internal nodes. It can be shown that it essentially depends on the passive network. In particular, we have that n_H is not larger than the rank of the corresponding matrix T.

Based on the algebraic properties of the matrix T, we can distinguish an important subclass of integrator-based Active-RC filters, i.e. state-space filters. Suppose that the matrix T is invertible, i.e. T^{-1} exists. Then we can rewrite (2.6) as

$$\begin{aligned} sX &= -T^{-1}GX - T^{-1}BU_{in} \\ U_{out} &= -CX - DU_{in} \end{aligned} \tag{2.12}$$

Let us denote

$$A_s = -T^{-1}G, \ B_s = -T^{-1}B, \ C_s = -C, \ D_s = -D \tag{2.13}$$

With this notation (2.12) takes the form

$$\begin{aligned} s\boldsymbol{X} &= \boldsymbol{A}_s\boldsymbol{X} + \boldsymbol{B}_s\boldsymbol{U}_{in} \\ \boldsymbol{U}_{out} &= \boldsymbol{C}_s\boldsymbol{X} + \boldsymbol{D}_s\boldsymbol{U}_{in} \end{aligned} \tag{2.14}$$

which is nothing else but the state-space description of the filter in Fig.1 with node voltages U_{xi}, i=1,...,n being the state variables. Thus, the state-space Active-RC filter is the one for which the matrix $\boldsymbol{T}$ is invertible which is a necessary and sufficient condition for the existence of the state matrices. For state-space filters we can apply many useful matrix transformations which can be used, for example, to perform efficient parameter optimization [13].

In practice Active-RC filters are mostly implemented in fully differential structures. Due to this we may assume that matrix entries in (2.5) can take both positive and negative values, which can be accomplished by cross-coupling of corresponding physical elements. More specifically, if the element, say R_{ij}, is cross-coupled (i.e. put between positive [negative] output of an amplifier and positive [negative] input of another one, see e.g. resistor R_1 in Fig.2), this reflects in equation (2.1) so that the appropriate term has the form $g_{ij}(-U_{xj})=-g_{ij}U_{xj}$, i.e. the original '-' from node voltage is moved into filter element, here g_{ij}. Obviously, the physical element remains positive. Negative value of the corresponding matrix entry is equivalent to cross-coupling. In case of single-ended implementation, negative elements must be realized using inverters.

The presented approach is primarily intended to be used as a basis for creating computer-aided design and optimization software. However, let us consider a simple example as an illustration how it can be used for filter design. Suppose that we want to synthesize an all-pole biquad filter, i.e. we intend to implement the following transfer function:

$$H(s)=\frac{H_o\omega_0^2}{s^2+\frac{\omega_0}{Q}s+\omega_0^2} \tag{2.15}$$

If one wants to develop minimal implementation (i.e. containing the smallest number of OPAMPs), two OPAMPs are needed, so we have n=2 in formulas (2.1)-(2.11). Assume that our filter has no input signal distribution with input signal injected to the first OPAMP through conductance g_b, i.e. $\boldsymbol{B}=[g_b\ 0]^T$, and no output summer with output signal taken from the second OPAMP, i.e. $\boldsymbol{C}$=[0 -1] and $\boldsymbol{D}$. Now we have to choose matrices $\boldsymbol{T}$ and $\boldsymbol{G}$ having in mind that in our case the transfer function of the filter is given by (2.10). More specifically, denominator of the transfer function is just $\det(s\boldsymbol{T}+\boldsymbol{G})$, while its numerator is $g_bA_{21}(s)$, where $A_{21}(s)=-[s\boldsymbol{T}+\boldsymbol{G}]_{21}$, i.e. element $_{21}$ of the matrix $s\boldsymbol{T}+\boldsymbol{G}$ (multiplied by -1). Although there are still many possibilities, the most straightforward choice is

$$\boldsymbol{T}=\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \qquad \boldsymbol{G}=\begin{bmatrix} 0 & -g_1 \\ g_2 & g_3 \end{bmatrix} \tag{2.16}$$

which gives all-pole second-order transfer function:

$$H(s)=\frac{g_bg_2}{s^2C_1C_2+sC_1g_3+g_1g_2}=\frac{g_bg_2/C_1C_2}{s^2+sg_3/C_2+g_1g_2/C_1C_2} \tag{2.17}$$

Corresponding filter topology (in fully differential structure) is shown in Fig.2. Obviously, we can obtain many more equivalent filter topologies that implement transfer function (2.15) by different choice of matrices $\boldsymbol{T}$, $\boldsymbol{G}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$.

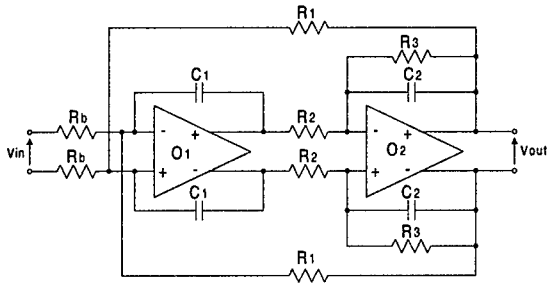


Fig.2. Active-RC biquad corresponding to matrices (2.16)

3. FILTERS WITH NON-IDEAL OPERATIONAL AMPLIFIERS

In this section we shall extend the model presented in Section 2 so as to take into account non-ideal behavior of operational amplifiers. In particular, we shall consider finite gain-bandwidth product of OPAMP as well as its non-zero output resistance.

In case of finite gain of OPAMP we can no longer assume that node voltages U_{zi}, i=1,...,n are equal to zero. Let us denote the gain of i-th operational amplifier O_i as $A_i(s)$. It follows from Fig.3 that i-th integrator of the filter can be described by the following equations:

$$y_{bi}(U_{in}-U_{zi})+\sum_{j=1}^{n} y_{ij}(U_{xj}-U_{zi})=0 \qquad i=1,...,n \tag{3.1}$$
$$U_{xi}=-U_{zi}A_i(s)$$

The corresponding equation for output summer is the following

$$g_o(U_{out}-U_{zo})+\sum_{j=1}^{n} g_{cj}(U_{xj}-U_{zo})+g_d(U_{in}-U_{zo})=0 \tag{3.2}$$
$$U_{out}=-U_{zo}A_o(s)$$

where $A_o(s)$ is the gain of the operational amplifier O_o and U_{zo} is the voltage at its input.

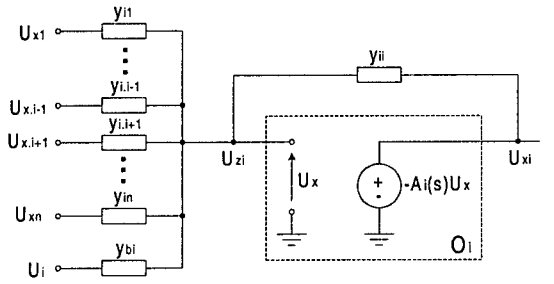


Fig.3. Diagram of i-th integrator of the filter in Fig.1 with finite gain OPAMP

Equations (3.1) and (3.2) can be rewritten in matrix form as follows

$$(sT+G+Y_A)X+BU_{in}=0$$
$$(1+C_A)U_{out}+CX+DU_{in}=0 \tag{3.3}$$

where

$$\boldsymbol{Y}_A = \begin{bmatrix} \dfrac{\sum_{j=1}^{n} \bar{y}_{1j} + \bar{y}_{b1}}{A_1(s)} & & 0 \\ & \ddots & \\ 0 & & \dfrac{\sum_{j=1}^{n} \bar{y}_{nj} + \bar{y}_{bn}}{A_n(s)} \end{bmatrix} \tag{3.4}$$

$$C_A = \frac{1+\sum_{j=1}^{n} \bar{c}_j + \bar{d}}{A_o(s)} \qquad (C_A=0 \text{ if there is no output summer}) \tag{3.5}$$

Here, $\bar{y}_{bi} = |g_{bi}| + s|C_{bi}|$, $\bar{y}_{ij} = |g_{ij}| + s|C_{ij}|$, $\bar{c}_i = |c_i|$, $i,j=1,...,n$, $\bar{d} = |d|$, which allows us to take into consideration the case when some of the matrices $\boldsymbol{T}$, $\boldsymbol{G}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ are negative. In such a case (which is equivalent to cross-coupling of the corresponding physical elements), negative sign of the element value corresponds in fact to the negative sign of appropriate node voltage U_{xj} (see discussion in Section 2). However, from the point of view of input node voltage U_{zi} element is still positive and we must take absolute value of negative elements to maintain correctness of equations. Using (3.3) we can calculate transfer function of the filter, which is:

$$H(s) = \frac{U_{in}(s)}{U_{out}(s)} = (1+C_A)^{-1}\left(\boldsymbol{C}(s\boldsymbol{T}+\boldsymbol{G}+\boldsymbol{Y}_A)^{-1}\boldsymbol{B}-\boldsymbol{D}\right) \tag{3.6}$$

Note that even if $A_i(s)$ is modeled using single pole approximation, matrix elements of $\boldsymbol{Y}_A$ are, in general, of second order in s. This makes it difficult to directly evaluate the transfer function formula (3.6). However, if for frequencies of operation of the filter we have $|A_i(s)|>>1$, $i=1,...,n$, $|A_o(s)|>>1$, which means that $\|\boldsymbol{Y}_A\|<<\|s\boldsymbol{T}+\boldsymbol{G}\|$ for any reasonable matrix norm, we can use the following approximation:

$$(\boldsymbol{I}+\boldsymbol{A})^{-1} \cong \boldsymbol{I}-\boldsymbol{A} \tag{3.7}$$

which is valid for any matrix $\boldsymbol{A}$ as far as $\|\boldsymbol{A}\|<<\|\boldsymbol{I}\|$ ($\boldsymbol{I}$ stands for an identity matrix). Using (3.7) we obtain the following formula for $H(s)$:

$$H(s) \cong (1+C_A)^{-1}\left(\boldsymbol{C}(s\boldsymbol{T}+\boldsymbol{G})^{-1}\boldsymbol{B}-\boldsymbol{D}\right) - (1+C_A)^{-1}\boldsymbol{C}(s\boldsymbol{T}+\boldsymbol{G})^{-1}\boldsymbol{Y}_A(s\boldsymbol{T}+\boldsymbol{G})^{-1}\boldsymbol{B} \tag{3.8}$$

which can be rewritten as

$$H(s) \cong (1+C_A)^{-1}H_0(s) - \Delta H(s) \tag{3.9}$$

where $H_0(s)$ is the nominal transfer function of the filter with ideal OPAMPs (eqn. (2.7)), and $\Delta H(s)$ is deviation from $H_0(s)$ due to finite OPAMP gains given by

$$\Delta H(s) = (1+C_A)^{-1}\boldsymbol{C}(s\boldsymbol{T}+\boldsymbol{G})^{-1}\boldsymbol{Y}_A(s\boldsymbol{T}+\boldsymbol{G})^{-1}\boldsymbol{B} \tag{3.10}$$

In general case, however, we have to evaluate formula (3.6) to obtain accurate results.

Now, we shall consider the effect of non-zero output impedance of the filter OPAMPs. Fig.4 shows i-th OPAMP of the filter, where $y_{oi}=1/z_{oi}$ denotes its output admittance, $\tilde{U}_{xi}$ and U_{xi} are the internal and external output voltages of the OPAMP, respectively. Since in practice we always have $|U_{zj}|<<|U_{xi}|$, $i,j=1,...,n$, we can write the following equations:

$$(\tilde{U}_{xi}-U_{xi})y_{oi} = U_{xi}\left(|g_{ci}| + \sum_{j=1}^{n}\bar{y}_{ji}\right), \qquad i=1,...,n \tag{3.11}$$

where we wrote $|g_{ci}|$ and $\bar{y}_{ji}$ for the same reason as in (3.4). Hence, we have

$$U_{xi} = \tilde{U}_{xi}\frac{y_{oi}}{y_{oi}+|g_{ci}|+\sum_{j=1}^{n}\bar{y}_{ji}}, \qquad i=1,...,n \tag{3.12}$$

This means that effective gain of the amplifier is reduced by the factor $\left(1+z_{oi}\left(|g_{ci}|+\sum_{j=1}^{n}\bar{y}_{ji}\right)\right)^{-1}$, which is, in general, frequency dependent. In order to take it into account in our previous formulas, we have to replace $A_i(s)$ in (3.4) by $\tilde{A}_i(s)$ given by

$$\tilde{A}_i(s)=\frac{A_i(s)}{1+z_{oi}\left(|g_{ci}|+\sum_{j=1}^{n}\bar{y}_{ji}\right)} \tag{3.13}$$

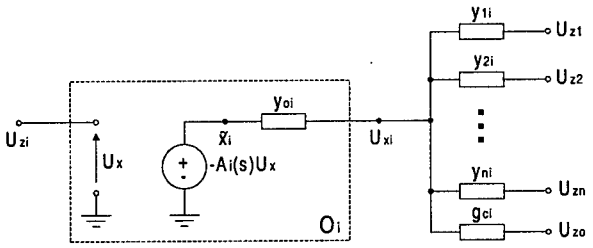


Fig.4. Diagram of i-th OPAMP of the filter in Fig.1 with non-zero output admittance

In a similar way, one can treat output impedance of O_o - OPAMP of the output summer of the filter. We have

$$\left(\tilde{U}_{out}-U_{out}\right)y_o=U_{out}|g_o|, \tag{3.14}$$

where $\tilde{U}_{out}$ and U_{out} are internal and external output voltages of O_o, while $y_o=1/z_o$ is its output admittance. Hence, we get

$$U_{out}=\tilde{U}_{out}\,y_o/(y_o+|g_o|), \tag{3.15}$$

which results in replacing $A_o(s)$ in (3.5) by $\tilde{A}_o(s)$:

$$\tilde{A}_o(s)=A(s)/(1+z_o|g_o|) \tag{3.16}$$

4. APPLICATION EXAMPLE

In this section we discuss an application example of the Active-RC filter model introduced in Sections 2 and 3 to obtain OPAMP design specifications for 3rd order low-pass 1dB Chebyshev filter for VDSL Analog Front End [7]. Design requirements for VDSL filters are very tough, especially with respect to noise and linearity. Here, we are only interested in transfer function distortion due to finite gain-bandwidth product and output resistance of filter OPAMPs. We shall consider leap-frog topology shown in Fig.5. Corner frequency of the filter is 12MHz. We assume for simplicity that all resistors in the filter are the same and equal to the common value R. The problem is to estimate required OPAMP gain-bandwidth product (GBW) and output resistance r_o so that distortion of the filter transfer function due to these non-idealities are within acceptable range.

Fig.6 shows ideal frequency response of the filter in Fig.5, and the response for GBW of OPAMP equal to 200MHz and r_o/R=0.5. We can observe both amplitude distortion (picking) ΔA and 3dB frequency error Δf_{3dB}. Figs.7 and 8 show ΔA and Δf_{3dB}, respectively, versus GBW of OPAMP, for different values of the ratio r_o/R. All the results have been obtained during a single run of the software implementing formulas presented in Sections 2

and 3. It is seen that Δf_{3dB} is not a big problem since its value is relatively small (several percent) and can be easily compensated by frequency calibration. However, ΔA is large and heavily dependent on both GBW and r_o/R.

Table 1 shows the value of GBW required to keep ΔA<2dB for different r_o/R ratios. The data in Table 1 represents trade-off between GBW and r_o of the OPAMP. Although we can relax requirements concerning GBW it is always at the expense of decreasing r_o/R ratio. It is not possible to keep this ratio small by just increasing filter resistor value R because this results in increasing filter noise, which is undesirable. Thus, we have an alternative: either try to obtain (very) large GBW or decrease OPAMP output resistance (this, on the other hand, influences OPAMP architecture - very small values of r_o can only be obtained using output buffer which, in turn degrades both circuit linearity and frequency performance). Discussion of this problem is, however, beyond the scope of this paper. Our goal was just to estimate GBW and r_o/R for which the transfer function distortion is under desired level.

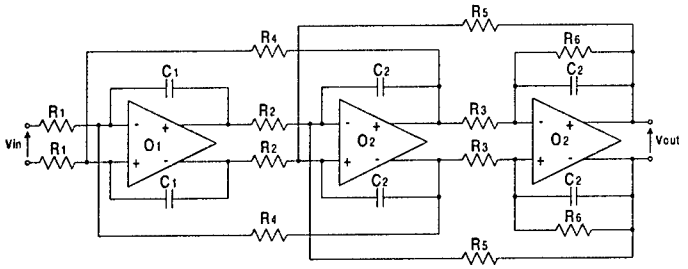


Fig.5. Fully differential 3rd order leap-frog Active-RC filter

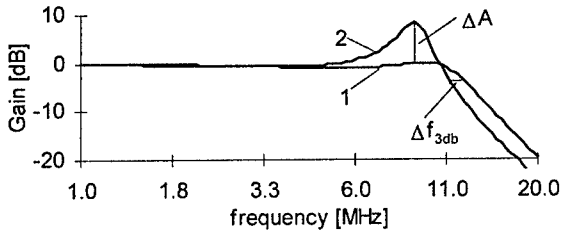


Fig.6. Transfer function distortion of the filter in Fig.5 due to finite OPAMP gain-bandwidth and output resistance; ideal response (1), actual response (2)

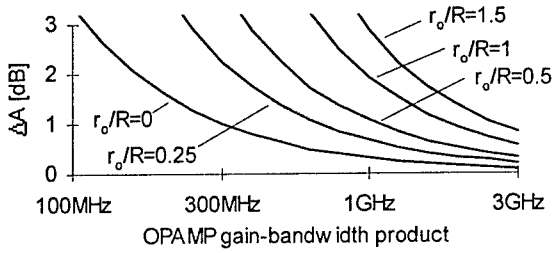


Fig.7. Amplitude distortion ΔA versus GBW of filter OPAMPs for different values of r_o/R ratio

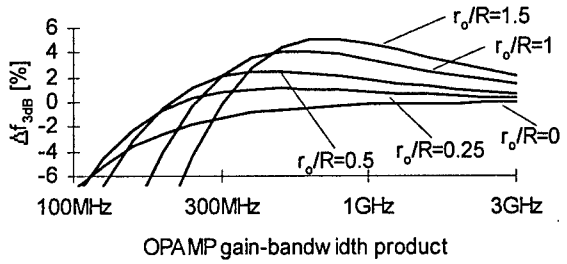


Fig.8. Corner frequency error Δf_{3dB} versus GBW of filter OPAMPs for different values of r_o/R ratio

Table 4.1

GBW of filter OPAMPs for which ΔA<2dB for the filter in Fig.5 with r_o/R ratio as a parameter

r_o/R	0	0.25	0.5	1.0	1.5
GBW [MHz]	165	350	550	970	1400

5. CONCLUSIONS

In the paper, a general structure of integrator-based Active-RC filter is introduced and analyzed using algebraic description. As a result, a matrix-based framework for creating efficient computer-aided analysis and design tools for Active-RC filters is developed. The extensions of the model are discussed that allow us to consider non-ideal OPAMPs including finite gain and bandwidth as well as non-zero output impedance. The goal of the future work is to develop - within the presented approach - tools for evaluating noise and nonlinear effects in Active-RC filters and use them in automated design/optimization system.

REFERENCES

[1] Schaumann R., Ghausi M.S., Laker K.R., *Design of Analog Filters, Passive, Active RC, and Switched Capacitor.* Englewood Cliff, NJ: Prentice-Hall, 1990.

[2] Schaumann R., Van Valkenburg M.A., *Design of Active Filters*, Oxford University Press, New York, 2001.

[3] Su K., *Analog Filters*, Kluwer Academic Publishers, 2002.

[4] Sun Y. (Editor), *Design of high frequency integrated analogue filters*, The Institution of Electrical Engineers, London, 2002.

[5] Lee S.-S., *Integration and System Design Trends of ADSL Analog Front Ends and Hybrid Line Interfaces*, in Proc. IEEE Custom Integrated Circuits Conf., pp.37-44, 2002.

[6] Weinberger H., Wiesbauer A., Clara M., Fleischhacker C., Potscher T., Seger B., *A 1.8V 450mW VDSL 4-Band Analog Front End IC in 0.18μm CMOS*, in Proc. IEEE Solid-State Circuits Conf. ISSCC, Vol.1, pp.326-471, 2002.

[7] Tan N., Caster F., Eichrodt C., George S.O., Horng B., Zhao J., *A Universal Quad AFE with Integrated Filters for VDSL, ADSL, and G.SHDSL*, in Proc. IEEE Custom Integrated Circuits Conf., pp.599-602, 2003.

[8] Jussila J., Halonen K., *A 1.5 V active RC filter for WCDMA applications*, in Proc. IEEE Int. Conf. Electronics Circuits. Syst. ICECS, Vol.1, pp.489-492, 1999.

[9] Khalil W., Chang T.-Y., Jiang X., Naqvi S.R., Nikjou B., Tseng J., *A highly integrated analog front-end for 3G*, IEEE J. Solid-State Circuits, Vol.38, pp.774-781, 2003.
[10] Frost C., Levy G., Allison B., *A CMOS 2 MHz self-calibrating bandpass filter for personal area networks*, in Proc. Int. Symp. Circuits Syst. ISCAS, Vol.1, pp.485-488, 2003.
[11] Wong C.S., *A 3-V GSM baseband transmitter*, IEEE J. Solid-State Circuits, Vol.34, pp.725-730, 1999.
[12] Harrison J., Weste N., *350MHz opamp-RC filter in 0.18μm CMOS*, Electron. Lett., Vol.38, pp.259-260, 2002.
[13] Koziel S., Szczepanski S., Schaumann R., *General Approach to Continuous-Time G_m-C Filters*, Int. J. Circuit Theory Appl., Vol.31, pp.361-383, July/Aug. 2003.

FILTRY AKTYWNE RC CZASU CIĄGŁEGO: MODEL OGÓLNY I JEGO ZASTOSOWANIA

Streszczenie

W pracy przedstawiono ogólny model filtru aktywnego RC opartego na wzmacniaczach odwracających fazę. Model analizowany jest przy użyciu równań macierzowych. Podano jawna formule określająca funkcje przenoszenia filtru w ogólnym przypadku. Przedstawiono rozszerzenia modelu pozwalające uwzględniać nieidealność wzmacniaczy operacyjnych, w tym skończone wzmocnienie i pasmo przenoszenia oraz niezerową rezystancję wyjściową. Zastosowanie opisu macierzowego umożliwia użycie modelu we wspomaganej komputerowo analizie i projektowaniu filtrów aktywnych RC. W pracy przedstawiono również przykład zastosowania modelu do szacowania wymagań projektowych dla dolnoprzepustowego filtru Czebyszewa trzeciego rzędu dla zastosowań w części analogowej modemu VDSL.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Slawomir Koziel, Stanislaw Szczepanski

Department of Microelectronics Systems, Gdansk University of Technology

NONLINEAR DISTORTION ANALYSIS AND OPTIMIZATION OF GENERAL OTA-C FILTERS

Summary

An analytical description of general OTA-C filters with nonlinear transconductors is introduced. A differential system that determines dynamics of a general structure of OTA-C filter is formulated. This allows us to carry out an effective and fast transient analysis of any OTA-C filter using standard numerical methods. The approach can be applied to investigate any nonlinear effects in filters. The accuracy of the proposed methods is confirmed by comparison with SPICE simulation. Example of application for performance optimization of 4^{th} order Chebyshev and Butterworth filters in cascade implementation is given.

1. INTRODUCTION

There has been a growing interest displayed in the design of continuous-time (CT) filters based on the transconductance-capacitor (OTA-C) technique for more than two decades [1], [2]. The operational transconductance amplifiers (OTAs) offer a higher bandwidth than their voltage-mode counterparts, can be easily tuned electronically, and have a better suitability for operating in reduced supply environment [3]-[5]. Due to this, high frequency integrated filters are mostly realized as the OTA-C ones [6].

Systems such as OTA-C filters are ideally designed to exhibit linear characteristics. However, their components – especially transconductors – are intrinsically nonlinear. Although there exist many approaches that aim at reducing nonlinear effects while dealing with practical design problems (see e.g. tutorial paper [7]), nonlinear distortion cannot be canceled out completely. Thus, it is important to estimate a degradation of filter performance caused by nonlinearities. In this paper we propose a simple and general method to perform a transient analysis of any OTA-C filter structure based on a matrix description and macromodeling of transconductors. This is a very fast and efficient procedure. Unlike the approaches based on Volterra series representation (e.g. [8]-[10]) or harmonic injection method [11], it is not restricted to handle weak nonlinearities only.

This work was supported in part by the State Scientific Research Committee, Poland, under Grant 4T11B01625

2. DYNAMICS OF NONLINEAR OTA-C FILTERS

Consider a general structure of OTA-C filter shown in Fig.1. The structure in Fig.1 contains n internal nodes denoted as x_i, $i=1,...,n$, n input transconductors G_{bi}, the set of internal feedback and feedforward transconductors G_{mij}, an output summer consisting of transconductors G_{ci} and G_o as well as a feedforward transconductor G_d. We assume that in general transconductors are not linear. All transconductors form *active network*, while input capacitors C_{bi}, $i=1,...,n$ and capacitors C_{ij}, $1\le i<j\le n$ form *passive network*. It is easily seen that any OTA-C filter is a particular case of the general structure in Fig.1.

We shall derive an analytical description of the considered structure in the time domain. In the following considerations we will denote the voltage at the i-th node x_i also by x_i, which will not lead to confusion. Symbols u_i, u_o will denote the input and output voltages, respectively. The rest of the necessary notation is introduced in Fig.2.

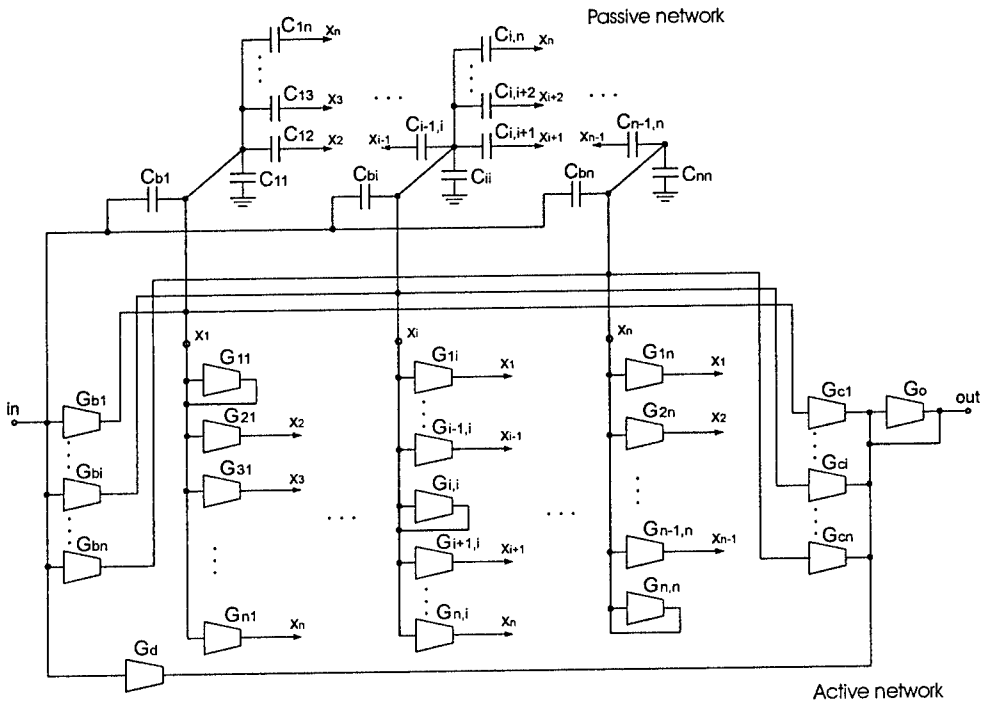


Fig.1. General structure of OTA-C filter

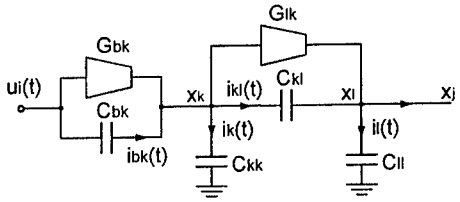


Fig.2. Part of the circuit in Fig.1 with the notation used to its analytical description

According to the notation introduced in Fig.2, the general structure of OTA-C filter in Fig.1 can be described by the following system of integral equations:

$$x_k(t) = \frac{1}{C_{kk}}\int_0^t i_k(s)ds + x_k(0), \tag{2.1}$$

$$x_k(t) - x_l(t) = \frac{1}{C_{kl}}\int_0^t i_{kl}(s)ds + x_k(0) - x_l(0), \quad (k<l) \tag{2.2}$$

$$u_i(t) - x_k(t) = \frac{1}{C_{bk}}\int_0^t i_{bk}(s)ds + u_i(0) - x_k(0), \tag{2.3}$$

$$-\sum_{q=1,\, q\neq k}^{n} i_{qk}(t) + i_k(t) - i_{bk}(t) = \sum_{q=1}^{n} G_{kq}\big(x_q(t)\big) + G_{bk}\big(u_i(t)\big), \tag{2.4}$$

$$u_o(t) = -G_o^{-1}\Big(\sum_{l=1}^{n} G_{cl}\big(x_l(t)\big) + G_d\big(u_i(t)\big)\Big), \tag{2.5}$$

where $k,l=1,...,n$, $k<l$. Note that G_{kl}, G_{bk}, G_{cl}, G_d and G_o, $k,l=1,2,...,n$ are in general nonlinear functions of their input variables. Define vectors $\boldsymbol{x}(t)$ and $\dot{\boldsymbol{x}}(t)$ as

$$\boldsymbol{x}(t) = [x_1(t) \ \cdots \ x_n(t)]^T, \quad \dot{\boldsymbol{x}}(t) = [\dot{x}_1(t) \ \cdots \ \dot{x}_n(t)]^T, \tag{2.6}$$

where symbol ^{T} denotes matrix transposition, and matrix $\boldsymbol{T_C}$ as

$$\boldsymbol{T_C} = \begin{bmatrix} C_{b1} + \sum_{j=1}^{n} C_{1j} & -C_{12} & \cdots & -C_{1n} \\ -C_{12} & C_{b2} + \sum_{j=1}^{n} C_{2j} & \cdots & -C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -C_{1n} & -C_{2n} & \cdots & C_{bn} + \sum_{j=1}^{n} C_{nj} \end{bmatrix}. \tag{2.7}$$

Using (2.6) and (2.7), the system (2.1)-(2.4) can be rewritten in equivalent differential form as:

$$\boldsymbol{T_C}\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \sum_{l=1}^{n} G_{1l}\big(x_l(t)\big) \\ \vdots \\ \sum_{l=1}^{n} G_{nl}\big(x_l(t)\big) \end{bmatrix} + \begin{bmatrix} G_{b1}\big(u_i(t)\big) + C_{b1}\dot{u}_i(t) \\ \vdots \\ G_{bn}\big(u_i(t)\big) + C_{bn}\dot{u}_i(t) \end{bmatrix}, \tag{2.8}$$

Let us now consider a special case, where all transconductors are linear, i.e. we have $G_{kl}(y) = g_{kl}y$, $G_{bk}(y) = g_{bk}y$, $G_{cl}(y) = g_{cl}y$, $G_d(y) = g_d y$ and $G_o(y) = g_o y$, $k,l = 1,2,...,n$. Define the following matrices:

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} g_{b1} + C_{b1}\dfrac{d}{dt} & \cdots & g_{bn} + C_{bn}\dfrac{d}{dt} \end{bmatrix}^T, \quad \boldsymbol{C} = [c_1 \ \cdots \ c_n], \quad \boldsymbol{D} = d \tag{2.9}$$

with $c_i = -g_{ci}/g_o$, $i=1,2,...,n$ and $d = -g_d/g_o$. Using this notation we can rewrite (2.8), (2.5) as follows:

$$\boldsymbol{T_C}\dot{\boldsymbol{x}}(t) = \boldsymbol{G}\boldsymbol{x}(t) + \boldsymbol{B}u_i(t), \qquad u_o(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}u_i(t), \tag{2.10}$$

or, in the domain of Laplace transform:

$$s\boldsymbol{T_C}\boldsymbol{X} = \boldsymbol{G}\boldsymbol{X} + \boldsymbol{B}u_i, \qquad u_o = \boldsymbol{C}\boldsymbol{X} + \boldsymbol{D}u_i, \tag{2.11}$$

in which $\boldsymbol{X}$ is the Laplace transform of the vector $\boldsymbol{x}$. System (2.11) is the state variable matrix description of the general structure of a voltage-mode OTA-C filter introduced

in [12]. Thus, system (2.11) is a particular (linear) case of the general (nonlinear) system describing a filter circuit in Fig.1.

Let us turn back to the general case. If the matrix $\boldsymbol{T}_C$ is invertible, equation (2.8) can be reformulated as

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{T}_C^{-1}\left(\begin{bmatrix}\sum_{l=1}^{n} G_{1l}(x_l(t))\\ \vdots \\ \sum_{l=1}^{n} G_{nl}(x_l(t))\end{bmatrix}+\begin{bmatrix}G_{b1}(u_i(t))+C_{b1}\dot{u}_i(t)\\ \vdots \\ G_{bn}(u_i(t))+C_{bn}\dot{u}_i(t)\end{bmatrix}\right). \tag{2.12}$$

The above assumption is very natural. In particular, it is satisfied if every internal node of the filter has a grounded capacitor. The problem of invertibility of matrix $\boldsymbol{T}_C$ has been thoroughly addressed in [12].

Denote the vector on the right-hand side of (2.12) by $f(u(t),x(t))$. Then we have

$$\dot{\boldsymbol{x}}(t)=f(u(t),\boldsymbol{x}(t)), \tag{2.13}$$

By endowing (14) with an initial condition

$$\boldsymbol{x}(0)=(x_{0.1},\ldots,x_{0.n}), \tag{2.14}$$

we arrive at a classical Cauchy problem which can be easily solved numerically.

The above model can be applied to calculate THD of any OTA-C filter for given input voltage frequency and amplitude. It is enough to integrate system (2.13), (2.14) with natural initial condition $x_{0.k}=0$ for $k=1,2,\ldots,n$ and calculate THD by definition, i.e.

$$THD=\sqrt{\sum_{n=2}^{\infty}C_n^2\Big/\sum_{n=1}^{\infty}C_n^2}\approx\sqrt{\sum_{n=2}^{N}C_n^2\Big/\sum_{n=1}^{N}C_n^2}, \tag{2.15}$$

where

$$C_n=\frac{2}{T}\left|\int_{-T/2}^{T/2}u_0(t)e^{-j2n\pi/T}dt\right|,\qquad n=1,2,\ldots \tag{2.16}$$

and N is integer chosen so as to make the approximation in (2.15) good enough (here $|z|$ denotes modulus of the complex number z). Obviously, coefficients C_n can be determined numerically using $u_0(t)$ obtained from (2.5).

3. VERIFICATION

In order to verify the accuracy of the proposed approach a comparison between theoretical results and SPICE simulations has been carried out. The numerical results have been obtained using 4th order Adams-Bashforth's [14] method to integrate differential equation (2.13) and three-point composite Newton-Cotes quadrature [14] to calculate coefficients C_n in (2.15).

For our comparison we use a simple differential-pair transconductor shown in Fig.3. The circuit was implemented in standard 0.35μm AMS technology and simulated using SPICE. The OTA macromodel parameters extracted from the simulations are: $i_{out}(v_{id}) = g_m v_{id} + g_{m3}(v_{id})^3 + g_{m5}(v_{id})^5 = 10^{-4}v_{id} - 4.2\cdot10^{-5}(v_{id})^3 - 3.4\cdot10^{-5}(v_{id})^5$ [μA] (v_{id} in volts; accuracy of the series better than 0.1% for $v_{id}<0.75$V). Transconductance of the circuit is $g_m=100$μA/V.

The OTA circuit in Fig.3 was used to implement the 3rd order Butterworth low-pass filter in a leap-frog (LF) structure shown in Fig.4 (note that the filter is realized in fully differential topology). The element values are: $C_1=2.37$pF, $C_2=2.11$pF, $C_3=0.79$pF; the OTAs in Fig.3 was used as active elements of the filter. 3dB frequency of the filter is 10MHz. Figs.5 and 6 show THD versus frequency with 0.3V input signal amplitude, and

THD versus input signal amplitude for 5MHz input frequency, respectively. The agreement between theoretical results (line) and SPICE simulation results (point) is very good. It should be emphasized that transient analysis performed by integrating (2.13) is, due to OTA macromodeling, orders of magnitude faster than the one carried out with transistor level simulation using SPICE.

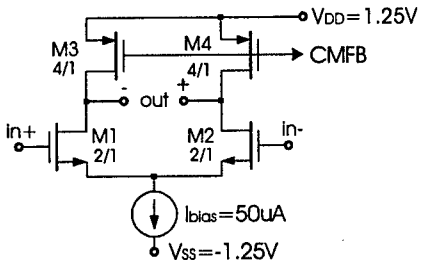


Fig.3. Simple differential-pair transconductor (Common-Mode Feedback circuit not shown)

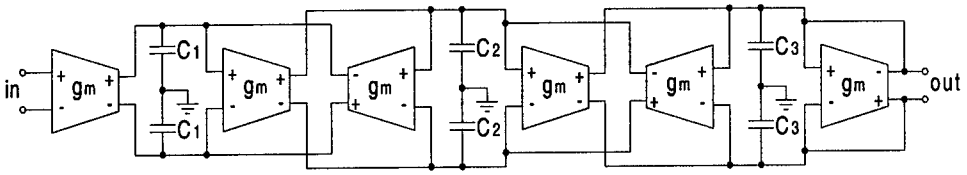


Fig.4. Diagram of 3rd order LF filter

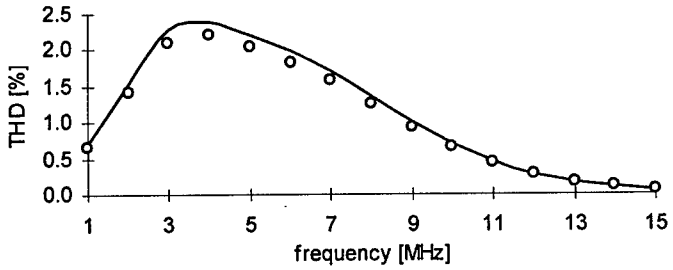


Fig.5. THD vs. frequency for filter in Fig.4 with 0.3V input amplitude: theoretical data (line), and simulation (points)

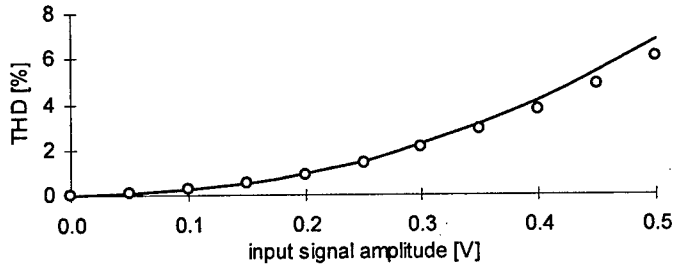


Fig.6. THD vs. input signal amplitude for filter in Fig.4 with 5MHz sine wave: theoretical data (line), and simulation (points)

4. APPLICATION EXAMPLE

As an application example consider performance optimization of two 4th order filters implemented in cascade topology: 1dB Chebyshev and Butterworth. Both filters use biquads shown in Fig.7 and transconductors realized with OTA circuits in Fig.3. Corner frequency of the filters is 10MHz. In the optimization process we have assumed two degrees of freedom. The first one is the biquad ordering (biquad A with normalized capacitors C_1 = 5.65pF, C_2 = 0.44pF, and biquad B with C_1 = 2.33pF, C_2 = 3.79pF for Chebyshev filter; biquad A with normalized capacitors C_1 = 1.96pF, C_2 = 1.15pF, and biquad B with C_1 = 0.82pF, C_2 = 2.78pF for Butterworth filter). The second are biquad gains K_A and K_B. Gains are adjusted by changing transconductance g_b; transconductance g_m=100μA/V is fixed; $g_b \in$ [70.7μA/V,141.4μA/V], which allows us to change K_A and K_B in the range of [−3dB,+3dB]. We assumed unity gain setting for the whole filter, i.e. K_A+K_B=0dB. Note that now nonlinearity parameters of input transcoductors g_b depend on transconductance value, i.e. $g_{m3} = g_{m3}(g_b)$, $g_{m5} = g_{m5}(g_b)$. In this case we modeled them using polynomial approximation. We omit the details for the sake of brevity.

Due to the fact that the approach presented in Section 2 allows us to perform very fast evaluation of nonlinearity and noise parameters of any filter, we were able to perform an exhaustive search through possible biquad configurations (*AB* and *BA*) and gain settings (using small grid of 0.06dB). Figs.8 and 9 show THD (at $0.6V_{pp}$ and 2MHz sinusoidal input) versus input biquad gain (i.e. K_A for *AB* biquad order and K_B for *BA*) for Chebyshev and Butterworth filter, respectively. We can observe that optimal (i.e. giving the best filter linearity) biquad ordering and gain distribution depends on filter transfer function (here, Chebyshev or Butterworth). Table 2.1 summarizes best configurations for both filters. Note that using the proposed approach we were able to obtain full knowledge about nonlinear distortion of the filters in question and choose the best setting that gives optimal linearity performance.

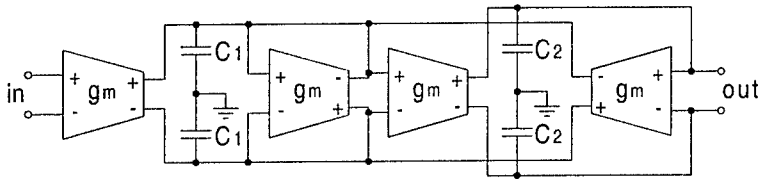


Fig.7. Diagram of OTA-C biquad

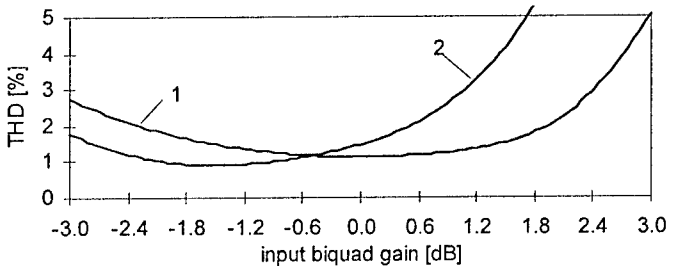


Fig.8. THD versus input biquad gain (K_A and K_B for *AB* and *BA* configuration, respectively) for 4th order cascade 1dB Chebyshev filter; configuration *AB* (1), and *BA* (2)

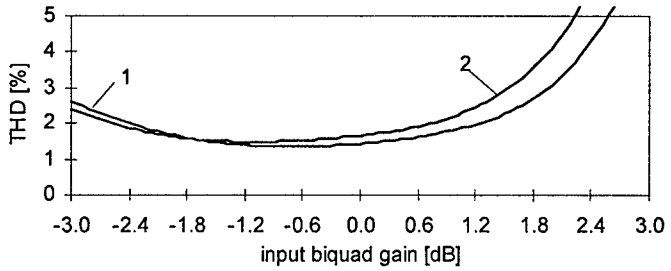


Fig.9. THD versus input biquad gain (K_A and K_B for *AB* and *BA* configuration, respectively) for 4th order cascade Butterworth filter; configuration *AB* (1), and *BA* (2)

Table 2.1

Optimal 4th order cascade Chebyshev and Butterworth configurations for best THD

Filter Transfer Function	Best THD Value [%]	Optimal Biquad Order	Configuration Biquad Gains [dB]	
1dB Chebyshev	0.90	*BA*	$K_A = 1.50$	$K_B = -1.50$
Butterworth	1.35	*AB*	$K_A = -0.72$	$K_B = 0.72$

5. CONCLUSIONS

A general description of OTA-C filters with nonlinear transconductors including derivation of a differential system describing time evolution of output signal for a general structure of OTA-C filter is formulated. The approach allows us to carry out fast transient analysis and investigate any nonlinear effects in arbitrary OTA-C filter. The accuracy of the theoretical results is confirmed by comparison to SPICE simulations. It follows that our results have a potential in applying to filter design and optimization, particularly creating effective tools for optimizing OTA-C filters with respect to nonlinear distortion. It is the matter of future work to extend the analysis to take into consideration noise of filters as well as develop automated system for optimization of OTA-C filters with respect to different performance criteria including dynamic range and signal to noise ratio.

REFERENCES

[1] Geiger R.L., Sánchez-Sinencio E., *Active filter design using operational transconductance amplifiers: A tutorial*, IEEE Circuit and Devices Mag., Vol.1, pp.20-32, 1985.

[2] Deliyannis T., Sun Y., Fidler J.K., *Continuous-time active filter design*, CRC Press, USA, 1999.

[3] Nauta B., Analog CMOS filters for very high frequencies, Kluwer Academic Publishers, 1993.

[4] Silva-Martinez J., Steyaert M., Sansen W., *High-Performance CMOS Continuous-Time Filters*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1993.

[5] Sun Y. (Editor), *Design of high frequency integrated analogue filters*, The Institution of Electrical Engineers, London, 2002.

[6] Tsividis Y.P., *Integrated continuous-time filter design – An overview*, IEEE J. Solid-State Circuits, vol. 29, pp. 166-176, Mar. 1994.

[7] Sánchez-Sinencio E., Silva-Martinez J., *CMOS transconductance amplifiers, architectures and active filters: a tutorial*, IEE Proc.-Circuits Dev. Syst., vol. 147, No. 1, pp. 3-12, Feb. 2000.
[8] Wambacq P., Sansen W., *Distortion Analysis of Analog Integrated Circuits*, Kluwer Academic Publishers, 1998.
[9] Szczepanski S., Schaumann R., *Nonlinearity Induced Distortion of the Transfer Function Shape in High-Order OTA-C Filters*, Analog Integrated Circuits and Signal Processing, vol.3, pp.143-151, 1992.
[10] Palaskas Y., Tsividis Y., *Dynamic Range Optimization of Weakly Nonlinear, Fully Balanced, OTA-C Filters With Power Dissipation Constraints*, IEEE Trans. Circuits Syst.-II, Vol.50, No.10, pp.714-727, Oct.2003.
[11] Mahattanakul J., Bunyakate C., *Harmonic injection method: a novel method for harmonic distortion analysis*, in Proc. Int. Symp. Circuits Syst. ISCAS, Vol.3, pp.85-88, 2001.
[12] Koziel S., Szczepanski S., Schaumann R., *General Approach to Continuous-Time OTA-C Filters*, Int. J. Circuit Theory Appl., Vol.31, pp.361-383, July/Aug. 2003.
[13] Brambilla A., Espinosa G., Montecchi F., Sánchez-Sinecio E., *Noise optimization in operational transconductance amplifier filters*, in Proc. Int. Symp. Circuits Syst. ISCAS, Vol.1, pp.118-121, 1989.
[14] Fortuna Z., Macukow B., Wasowski J., *Metody Numeryczne*, Wydawnictwa Naukowo Techniczne, Warszawa, 1993.

ANALIZA I OPTYMALIZACJA ZNIEKSZTAŁCEŃ NIELINIOWYCH W OGÓLNYCH FILTRACH OTA-C

Streszczenie

W pracy przedstawiono opis analityczny ogólnej struktury filtru OTA-C zawierającego nieliniowe elementy aktywne. Wyprowadzono nieliniowy układ równań różniczkowych zwyczajnych opisujący dynamikę filtru. Opis ten pozwala, przy użyciu standardowych metod numerycznych, na przeprowadzenie efektywnej analizy czasowej dowolnego filtru OTA-C. Proponowane podejście może być stosowane do badania dowolnych efektów nieliniowych w filtrach tej klasy. Dokładność metody została zweryfikowana poprzez porównanie z wynikami otrzymanymi za pomocą symulatora SPICE. W pracy przedstawiono również przykład zastosowania opisywanej metody do optymalizacji zniekształceń nieliniowych w dolnoprzepustowym filtrze 4-go rzędu realizującym równomiernie falistą charakterystykę przenoszenia.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Bogdan Pankiewicz, Mariusz Solecki

**Katedra Systemów Mikroelektronicznych,
Wydział Elektroniki Telekomunikacji i Informatyki Politechniki Gdańskiej**

REZYSTOR AKTYWNY CMOS ORAZ JEGO ZASTOSOWANIE DO BUDOWY WZMACNIACZA TRANSKONDUKTANCYJNEGO

Streszczenie

Rezystory aktywne są szeroko stosowane w układach analogowych. Jednym z bardzo ważnych zastosowań jest ich użycie jako elementu linearyzującego stałoprądowe charakterystyki przejściowe wzmacniacza różnicowego wykonanego w postaci pary tranzystorów MOS. W niniejszym artykule przedstawiono w pełni różnicowy, aktywny rezystor MOS. Składa się on z dwóch tranzystorów MOS oraz dwóch nieuziemionych źródeł napięciowych. Jako wykorzystanie rezystora przedstawiono zlinearyzowany wzmacniacz transkonduktancyjny OTA (ang. *Operational Transconductance Amplifier*). Wymienione układy zostały zaprojektowane jak również fizycznie wykonane w technologii AMS CMOS 0.8um. W pracy przedstawiono opis teoretyczny, wyniki symulacji komputerowych jak również wyniki pomiarowe układu prototypowego.

1. WSTĘP

Rezystory aktywne są szeroko stosowane w analogowych układach scalonych. Jednym z głównych zastosowań jest linearyzacja charakterystyk przejściowych wzmacniaczy elektrycznych, np. par różnicowych [1,2]. W tym artykule przedstawiono całkowicie różnicowy rezystor MOS złożony z dwóch tranzystorów MOS oraz dwóch nieuziemionych źródeł napięciowych. Jako zastosowanie zaprezentowano różnicowy wzmacniacz transkonduktancyjny przewidziany dla technologii CMOS.

2. RÓŻNICOWY REZYSTOR MOS

Układ rezystora aktywnego przedstawiony jest na rys. 1. Składa się on z tranzystorów MOS M12 i M21 oraz dwóch źródeł napięciowych, każde z nich o tej samej wydajności napięciowej równej v_{GS}. Układ jest podobny do przedstawionego w publikacji [2], różnica polega na tym, że w zaproponowanym rozwiązaniu tranzystory pracują z zakresie omowym i odwrotnym omowym zamiast wyłącznie omowego. Każdy z tranzystorów jest umieszczony na oddzielnej wyspie i ma zwarte wyprowadzenia podłoża i źródła. Jeśli założymy że napięcie v jest dodatnie, wówczas tranzystor M12 pracuje w zakresie omowym

natomiast M21 pozostaje w zakresie omowym odwrotnym. Stosując standardowy opis kwadratowy charakterystyk tranzystorów MOS, prądy drenów tranzystorów można przedstawić w postaci [2]:

$$i_{D12} = 2K\left(v_{GS} - V_{T0n} - \frac{v}{2}\right)v \tag{2.1}$$

$$i_{D21} = -2K\left(v_{GS} - V_{T2} + \frac{v}{2}\right)v \tag{2.2}$$

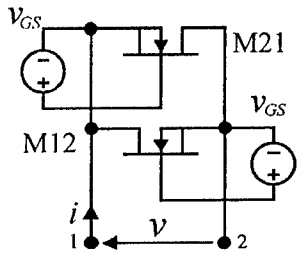
gdzie: $K=0.5\mu_n C_{ox}W/L$ jest współczynnikiem transkonduktancyjnym identycznych elementów M12 i M21, V_{T0n} to napięcie progowe przy zerowym napięciu podłoże – źródło ($V_{BS}=0V$). Napięcie progowe tranzystora M21 można przedstawić równaniem $V_{T2} = V_{T0n} - \gamma\left(\sqrt{\phi - v} - \sqrt{\phi}\right)$, gdzie γ jest parametrem podłożowym natomiast ϕ jest potencjałem powierzchniowym. Jeśli prąd płynący przez aktywny rezystor zdefiniujemy jako różnicę prądów drenów elementów M12 i M21, wówczas otrzymamy jego wartość równą:

$$\begin{aligned} i &= i_{D12} - i_{D21} = \\ &= 2Kv\left[2\left(v_{GS} - V_{T0n}\right) + \gamma\sqrt{\phi} - \gamma\sqrt{\phi - v}\right] \end{aligned} \tag{2.3}$$

Spełnienie nierówności $2\left(v_{GS} - V_{T0n}\right) >> \gamma\sqrt{\phi} - \gamma\sqrt{\phi - v}$ powoduje powstanie liniowej zależności napięciowo-prądowej a wypadkowa rezystancja jest równa:

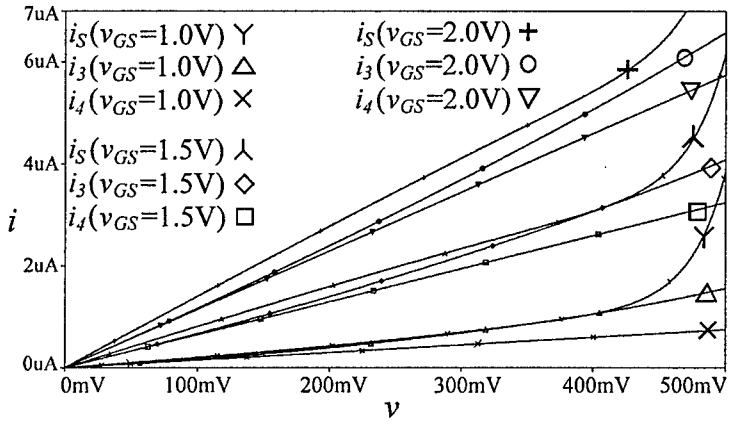
$$r = \frac{1}{4K\left(v_{GS} - V_{T0n}\right)} \tag{2.4}$$

Zakres napięcia v dla którego zachowany jest liniowy charakter rezystancji (2.4) jest ograniczony pracą jednego z tranzystorów (M12 lub M21 w zależności od polaryzacji źródła v) w zakresie odwrotnym omowym. Aby zapewnić niskie wartości prądów podłożowych napięcie v powinno być ograniczone do zakresu około ±0.5V.



Rys. 1. Różnicowy, aktywny rezystor MOS.

Rys. 2 przedstawia charakterystyki napięciowo prądowe rezystora z rys. 1 wykonane z użyciem symulacji komputerowych. Krzywe oznaczone i_S otrzymano przy pomocy symulatora PSPICE oraz modeli BSIM v3.1 dla technologii AMS 0.8um. Prądy oznaczone symbolami i_3 oraz i_4 przedstawiają odpowiednio równanie (2.3) oraz to samo równanie z pominiętym czynnikiem nieliniowym, tj. przy założeniu że $\gamma\sqrt{\phi} - \gamma\sqrt{\phi - v} = 0$.

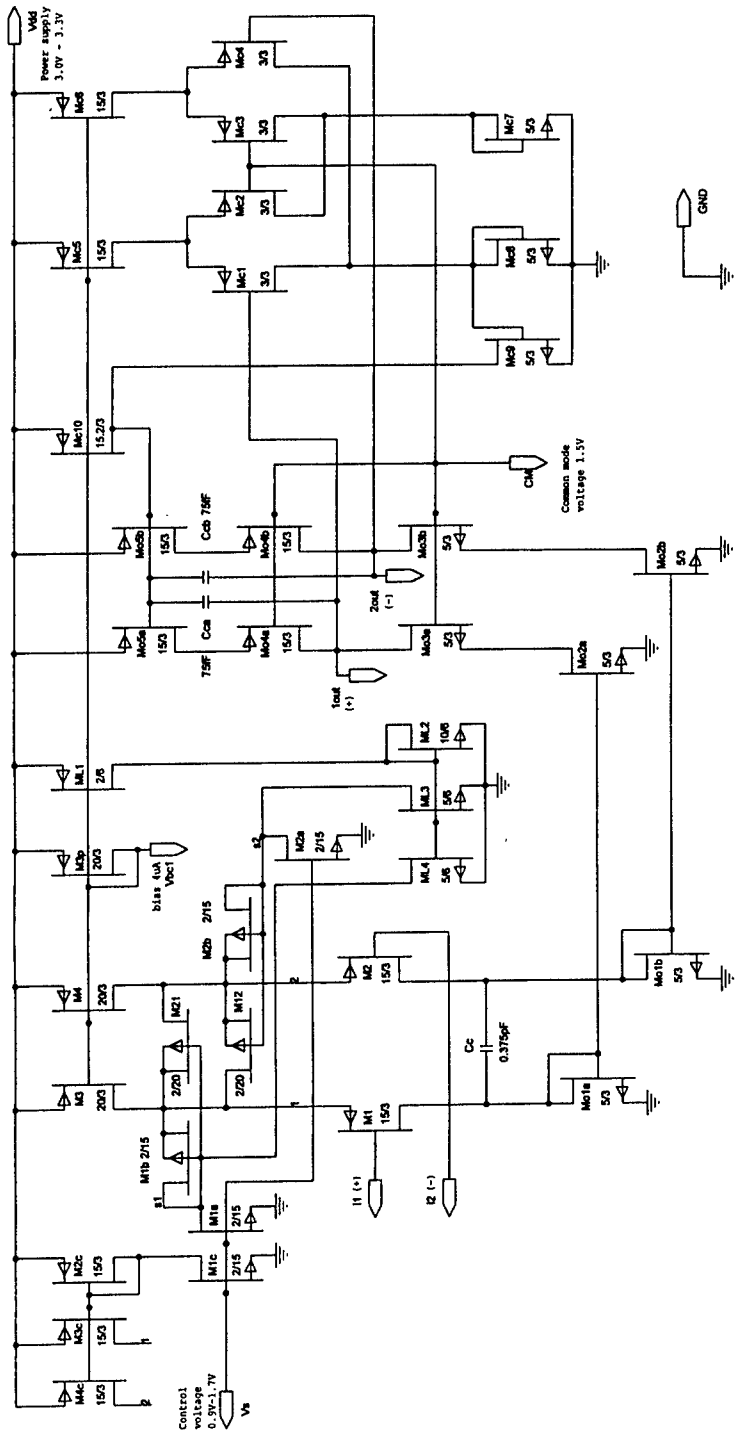


Rys. 2. Napięciowo – prądowe charakterystyki rezystora z rys. 1 otrzymane poprzez symulacje komputerowe. Oznaczenia prądów: i_S – symulacja PSPICE model BSIM v3.1, i_3 – wg równania (2.3), i_4 – jak i_3 ale przy założeniu, że $\gamma\sqrt{\phi} - \gamma\sqrt{\phi - v} = 0$.

3. OPERACYJNY WZMACNIACZ TRANSKONDUKTANCYJNY

Schemat wzmacniacza transkonduktancyjnego przedstawiony jest na rys. 3. Tranzystory M1 i M2 tworzą zlinearyzowaną parę różnicową MOS. Jako rezystor linearyzujący wykorzystano układ przedstawiony wcześniej na rys. 1, stąd wynikowa wartość transkonduktancji pary różnicowej jest w przybliżeniu równa odwrotności rezystancji danej wzorem (2.4). Napięcia v_{GS} niezbędne do polaryzacji tranzystorów M21 i M12 uzyskano z użyciem tranzystorów M1b oraz M2b w połączeniu diodowym, przez które płynie prąd regulowany napięciem bramkowym V_S tranzystorów M1a, M2a tworzących źródła prądowe.

Elementy ML1-ML4 zostały wprowadzone w celu ograniczenia minimalnej wartości transkonduktancji. Transkonduktancja stopnia wejściowego wzmacniacza jest odwrotnością wyrażenia (2.4), gdyby więc ustalono $V_{GS}=V_{T0n}$, wartość transkonduktancji byłaby równa zeru. Taka możliwość jest często niewygodna dla układów automatycznego dostrajania filtrów. Z tego względu wprowadzone zostały dodatkowe, stałe prądy polaryzujące wytwarzane w drenach tranzystorów ML3 oraz ML4, które w efekcie ograniczają od dołu minimalną osiąganą wartość napięcia bramka – źródło tranzystorów M12 i M21.



Rys. 3. Wzmacniacz transkonduktancyjny CMOS z wykorzystaniem rezystora aktywnego z rys. 1.

Tranzystory Mo1a-Mo5a oraz Mo1b-Mo5b tworzą układ wyjściowego lustra prądowego w konfiguracji kaskody. Kondensator CC został wprowadzony w celu kompensacji charakterystyk częstotliwościowych wzmacniacza. Ze względu na stosowanie luster prądowych oraz możliwość uzyskania niskich wartości transkonduktancji, przedstawiony wzmacniacz transkonduktancyjny przewidziany jest do pracy w zakresie niskich i średnich częstotliwości (do kilkunastu MHz).

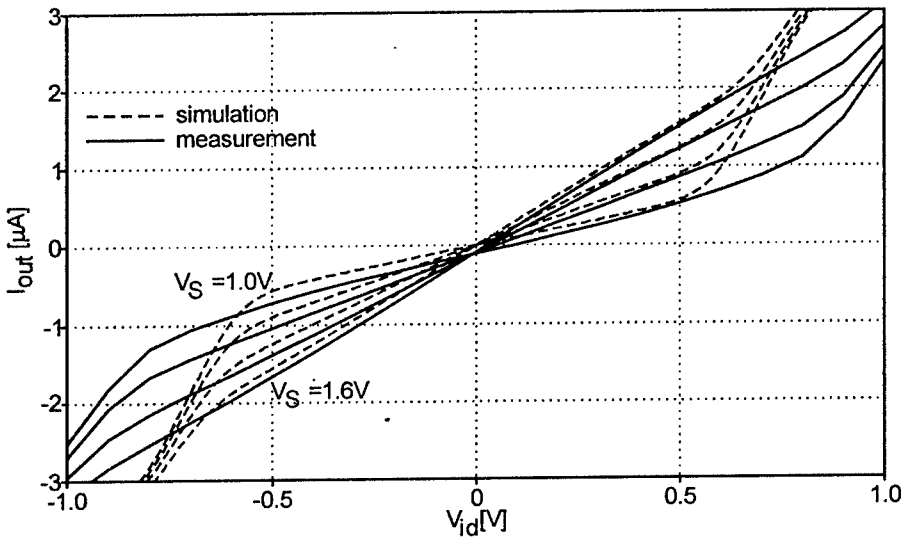
Tranzystory Mc1-Mc10 tworzą układ stabilizacji składowej stałej sygnału wyjściowego CMFB (ang. Common Mode Feed Back). Wartość średnia napięć na wyprowadzeniach 1out oraz 2out jest stabilizowana na poziomie napięcia podawanego na wyprowadzenie CM.

Prototypowy układ scalony został wykonany w technologii 0.8um CYE AMS CMOS. Projekt topografii wykonano przy użyciu pakietu CADENCE v.4.4.3. W układzie tym zawarto m. in. przedstawiony powyżej wzmacniacz transkonduktancyjny. Wyniki pomiarów i symulacji przedstawione zostały na rys. 4 oraz w tablicy 3.1. Zakres napięcia wejściowego dla którego otrzymano liniową charakterystykę przejściową jest podobny jak dla rezystora aktywnego z rys. 1 i wynosi około ±0.5V. Niestety rzeczywisty wzmacniacz transkonduktancyjny wykazuje dużą wartość wejściowego napięcia niezrównoważenia, dochodzącą prawie do 100mV. Układy CMOS charakteryzują się dużymi napięciami niezrównoważenia i dlatego często stosowane są specjalne techniki kompensujące [3]. Stabilizacja wartości średniej napięcia wyjściowego jest bardzo dobra, odchyłka od zadanej wartości 1.5V nie przekracza 4.1mV. Błąd ten jest spowodowany głównie poprzez niedopasowanie (nieidentyczność) elementów układu CMFB.

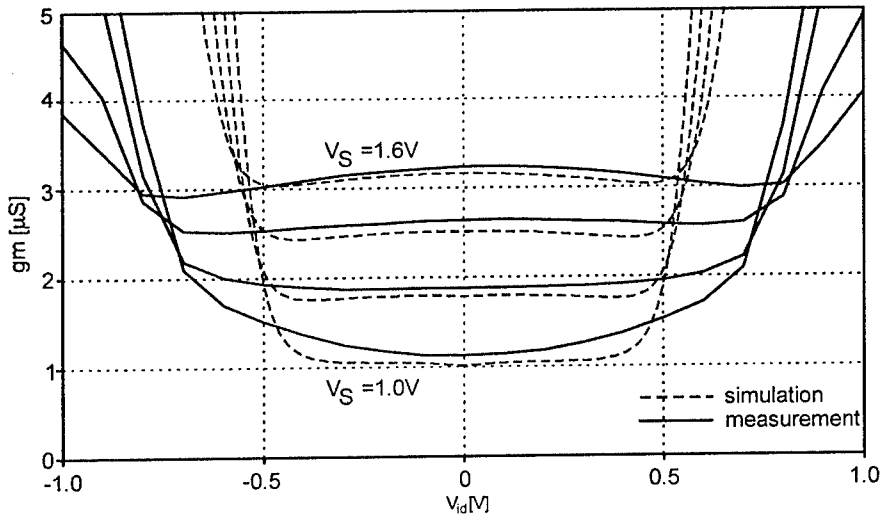
Tablica 3.1

Wybrane parametry wzmacniacza transkonduktancyjnego dla napięcia zasilającego VDD=3.3V.

Parametr	**Jednostka**	**Wyniki symulacji**	**Wartość zmierzona**
Zakres transkonduktancji dla Vs=1 do 1.6V	[μS]	1.02-3.15	1.1-3.25
Wejściowe napięcie niezrównoważenia dla Vs=1.3V, dla 3 układów scalonych	[mV]	---	7.74-97.1
Wspólne napięcie wyjściowe	[V]	1.4950-1.4951	1.459-1.471
Pobór mocy	[mW]	0.0997	---
Różnicowa pojemność wejściowa	[fF]	15,41	---
Różnicowa pojemność wyjściowa	[fF]	44.83	---
Pasmo 3dB prądu wyjściowego	[MHz]	29.5	---



(a)



(b)

Rys. 4. Charakterystyki przejściowe wzmacniacza transkonduktancyjnego z rys. 3.
(a) prąd wyjściowy płynący przez zwarte wyprowadzenia 1out i 2out $i_{OUT}=f(v_{id})$,
(b) transkonduktancja $gm = \partial i_{OUT}/\partial v_{id} = f(v_{id})$.

4. WNIOSKI KOŃCOWE

W artykule przedstawiono nową koncepcję różnicowego rezystora aktywnego CMOS. Składa się on z dwóch tranzystorów MOS, pracujących w zakresach omowym i omowym odwrotnym. Ze względu na duży wpływ efektu podłożowego każdy z tranzystorów powinien znajdować się na osobnej wyspie. Zaproponowaną strukturę rezystora wykorzystano do budowy zlinearyzowanej pary różnicowej MOS a następnie całkowicie różnicowego wzmacniacza transkonduktancyjnego. Liniowy zakres napięcia wejściowego jest ograniczony do około ±0.5V. Przedstawiono wyniki symulacji komputerowych (PSPICE, model BSIM v3.1) oraz pomiarów układu prototypowego, są one zbieżne. Niestety rzeczywisty wzmacniacz ma dużą wartość wejściowego napięcia niezrównoważenia i w przyszłości należy rozważyć wykonanie lepszego projektu topografii oraz zastosowanie układów kompensujących.

BIBLIOGRAFIA

[1] R. R. Torrance, T. R. Viswanathan and J. V. Hanson, "CMOS voltage to current transducers", IEEE Transactions on Circuits and Systems, Vol. CAS-32, No 11, pp. 1097-1104, November 1985.

[2] Geiger R.L., Allen P. E. and Strader N. R., "VLSI design techniques for analog and digital circuits", McGraw-Hill 1990, pp.178-179, 308-309.

[3] ispPAC^{TM} 10 technical data sheet, Lattice Semiconductor Corporation, Oregon 97124, USA, March 2000.

AN ACTIVE MOS RESISTOR AND ITS APPLICATION TO OPERATIONAL TRANSCONDUCTANCE AMPLIFIER

Summary

Active MOS resistors are widely used especially in analogue circuits. One of the very important application of such element is the linearisation of transfer characteristics of transconductance amplifiers built of differential MOS pairs [1,2]. In this paper a differential active MOS resistor is developed. It consists of two MOS devices and two floating voltage sources. As an application, a fully differential operational transconductance amplifier is presented (OTA). The proposed OTA circuit was designed and fabricated using the 0.8μm CMOS process (CYE AMS). Theoretical, simulation and measurement results are presented and compared.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Władysław Szcześniak, Piotr Szcześniak*

Katedra Systemów Geoinformatycznych, Politechnika Gdańska
***Ośrodek Badawczo-Rozwojowy, Centrum Techniki Morskiej**

ALGORYTMICZNE METODY REDUKCJI POBORU MOCY W CYFROWYCH UKŁADACH CMOS

Streszczenie

W pracy przedstawiono wybrane możliwości zredukowania poboru mocy układów cyfrowych CMOS z wykorzystaniem heurystycznych algorytmów szeregowania zadań w procesie syntezy wysokiego poziomu. Zaprezentowano przy tym trzy opracowane algorytmy IIOI (ang. *Inserting Idle Operation with Interchanging*) – wstawianie pustych operacji z wymianą jednostek funkcjonalnych, MAREL (ang. MAximal RELativity – maksymalne pokrewieństwo) i UNILO (ang. *UNIform LOad* – jednakowe obciążenie), dla których utworzono odpowiednie programy pozwalające praktycznie uzyskać redukcję mocy projektowanych układów CMOS. Przeprowadzone (przy pomocy standardowych benchmarków ISCAS'85 i ISCAS'89) badania testowe wykazały znaczną redukcję mocy (do ok. 40%) bez pogarszania wydajności systemu i zwiększania jego czasu przetwarzania.

1. WSTĘP

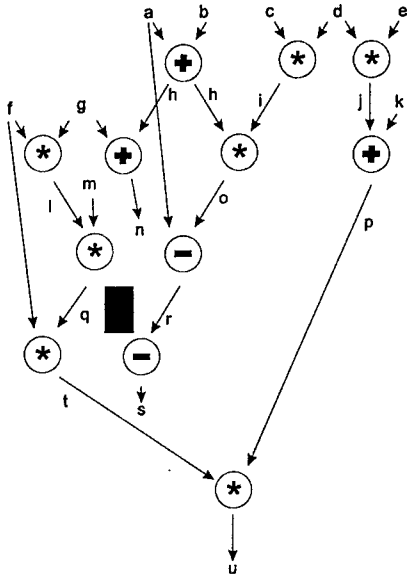
Projektowanie układów VLSI z uwzględnieniem kryteriów redukcji poboru mocy jest określane w literaturze przedmiotu jako jeden z najważniejszych problemów dotyczących przyrządów półprzewodnikowych i układów scalonych następnej generacji [1]. Związane jest to przede wszystkim z gwałtownym wzrostem zapotrzebowania na systemy przenośne, które są zasilane ze źródeł prądu stałego o skończonej pojemności. Ponadto zmniejszenie w układzie scalonym gęstości rozpraszanego ciepła ma istotny wpływ na zwiększenie jego niezawodności, a także decyduje o kosztach jego wytwarzania i eksploatacji, z uwagi na zastosowanie tańszych obudów i systemów chłodzenia. Czynniki te wymuszają rozwój badań nad redukcją poboru mocy przez układy scalone VLSI CMOS. Aktualnie dostępna literatura z tej dziedziny (za ostatnie dziesięć lat) zawiera kilkaset prac, opisujących różnorodne zastosowania i technologie [1-5].

W celu uzyskania redukcji mocy rozpraszanej na ogół stosuje się w układzie scalonym obniżanie napięć zasilających wybrane jednostki funkcjonalne, które wchodzą w skład projektowanego systemu elektronicznego, a dzieje się to bez pogarszania wydajności (przepustowości) całego systemu.

Podejście zaproponowane w pracy uwzględnia minimalizację mocy pobieranej ze źródła (P_b) z jednoczesnym uzyskaniem założonych parametrów czasowych projektowanego systemu elektronicznego. Realizowane jest ono na etapie syntezy logicznej układu.

2. REPREZENTACJA DANYCH WEJŚCIOWYCH OPISANYCH ALGORYTMEM

W celu uzyskania modelu danych, który prezentuje algorytmy podlegające syntezie wysokiego poziomu, stosuje się graf przepływu danych [2] (ang. *Data Flow Graph–DFG*). Przykład takiego grafu przedstawiono na rys. 1.



Rys. 1. Przykład grafu *DVG(V, E)*

Graf *DFG(V,E)* jest grafem skierowanym, w którym zbiór wierzchołków $V=\{v_i\}$ reprezentuje operacje. Zbiór krawędzi $E=\{e_{lm}\}$ reprezentuje wartości, przekazywane pomiędzy operacjami. Operacje reprezentują logiczne działania, które w rzeczywistym układzie są wykonywane przez zbiór jednostek funkcjonalnych $\{fu_i\}$ o ograniczonej liczności.

3. OPIS PROBLEMU

W opracowanej metodzie wprowadzono podejście mające zastosowanie w cyfrowych układach CMOS, dla których moc dynamiczna pobierana ze źródła zasilającego jest dana wzorem [3], [4]:

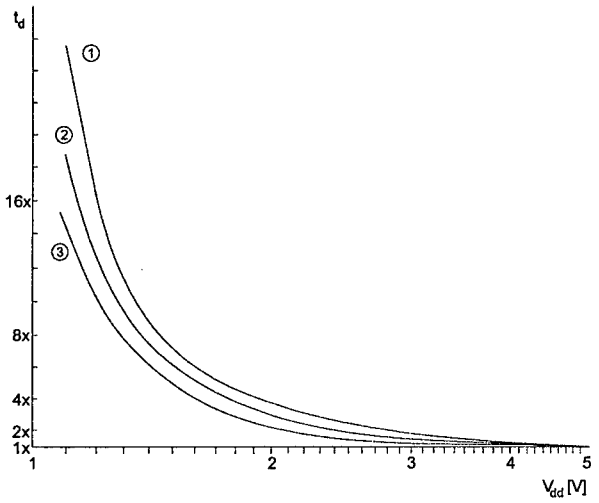
$$P_{Li} = \frac{a_i}{2} C_{Li} V_{ddi}^2 f_{clk} \tag{3.1}$$

gdzie: a_i – współczynnik aktywności,
C_{Li} – obciążenie pojemnościowe,
V_{ddi} – napięcie zasilania,
f_{clk} – częstotliwość przełączania.

Obniżenie napięcia V_{ddi} określonej jednostki funkcjonalnej powoduje zatem zredukowanie mocy pobieranej ze źródła zasilającego przy jednoczesnym wydłużeniu jej nominalnego czasu opóźnienia zgodnie z zależnością [3], [4]:

$$t_{dNi} \cong \frac{C_{Li} V_{ddi}}{K\left(V_{ddi} - V_t\right)^{\alpha}} \tag{3.2}$$

gdzie: K – stała technologiczna,
α – stała technologiczna, $\alpha \in (1,2)$,
V_t – napięcie progowe.



Rys. 2. Unormowany czas opóźnienia trzech różnych jednostek funkcjonalnych w funkcji napięcia zasilającego (na podstawie symulacji i pomiarów przedstawionych w pracy [5]).

Z kolei wartość napięcia progowego V_t zależy od technologii wykonania układu MOS, napięcia V_{dd} i aktualnego zakresu pracy tranzystorów. Przykładowy wpływ napięcia zasilania V_{dd} na unormowany czas opóźnienia t_d przedstawiono na rys. 2.

Dla rzeczywistych potrzeb projektowania można dobrać takie pary wartości napięć $\left(V_{ddi}^{A}, V_t^{A}\right)$ i $\left(V_{ddi}^{B}, V_t^{B}\right)$, dla których odpowiadające im opóźnienia nominalne będą określone jako:

$$t_{dN}^{A} = \frac{1}{2} t_{dN}^{B} \tag{3.3}$$

Oznacza to dwukrotnie wolniejsze działanie jednostki funkcjonalnej fu_A w stosunku do fu_B. W zamian za to, w wyniku obniżenia napięcia V_{ddA}, zmniejszeniu ulega także pobór mocy dynamicznej. Wybierając zatem odpowiednio jednostki funkcjonalne układu, które mogą pracować z podwójnie zmniejszoną częstotliwością przełączania bez pogarszania parametrów czasowych całego układu, uzyskujemy redukcję jego poboru mocy.

Zadaniem prezentowanych algorytmów jest taki przydział operacji do jednostek funkcjonalnych, aby maksymalna ich liczba mogła zostać spowolniona bez pogorszenia wydajności całego układu. W każdym z opracowanych algorytmów szeregowania zadań zakłada się wykorzystywanie spowolnionych jednostek funkcjonalnych, pracujących 2^n, ($n \in \{1,2,3\}$) razy wolniej. Dobór stopnia spowolnienia n zależy od założeń projektowych (maksymalnych dopuszczalnych wydłużeń czasu przetwarzania, struktury grafu *DFG*).

Wynikiem opracowanych algorytmów redukcji poboru mocy na etapie syntezy wysokiego poziomu jest graf uszeregowania zadań (*SG* – ang. *Scheduling Graph*), w którym każdy wierzchołek *DFG* jest przyporządkowany do odpowiedniej jednostki funkcjonalnej i taktu zegara w taki sposób, że wszystkie zależności pomiędzy operacjami są spełnione. Każda kolumna *SG* reprezentuje jedną jednostkę funkcjonalną, a każdy wiersz – kolejny takt zegara.

4. OPIS ALGORYTMÓW SZEREGOWANIA ZADAŃ

Poniżej omówiono trzy algorytmy szeregowania zadań dla potrzeb redukcji mocy w układach cyfrowych CMOS. Jako pierwszy opisano algorytm IIOI, na którym bazują pozostałe dwie modyfikacje. Dokonuje on wstawiania pustych operacji w celu spowolnienia wybranych jednostek funkcjonalnych, dzięki czemu (na podstawie zależności omówionych we wprowadzeniu) możliwe jest zredukowanie napięcia zasilania tych jednostek funkcjonalnych. Prowadzi to do redukcji mocy pobieranej przez układ cyfrowy.

4.1. Algorytm szeregowania zadań IIOI i jego modyfikacje

Zastosowanie podstawowego algorytmu szeregowania zadań pozwala na wydłużenie czasu realizacji operacji na określonej jednostce funkcjonalnej fu_i w sąsiedztwie luk czasowych. Dodatkowe dopuszczenie możliwości wymiany operacji przydzielonych do równolegle pracujących jednostek funkcjonalnych pozwala na zredukowanie opóźnień dostępności wyników pośrednich, co zapobiega nadmiernemu wydłużaniu grafu *SG*.

Algorytm IIOI podzielony jest na dwie części:

- faza wstępna – przygotowanie szeregowania ASAP (ang. as soon possible),
- faza właściwa – dokonanie przemieszczeń i spowolnień w grafie *SG* dla redukcji poboru mocy.

4.1.1. Algorytm ASAP

Pierwszą fazą prezentowanych algorytmów jest algorytm ASAP. W znacznej części jest on wspólny dla algorytmów IIOI, MAREL i UNILO. Działa on na zasadzie przydziału operacji do pierwszej wolnej jednostki funkcjonalnej tak szybko, jak jest to możliwe.

Z punktu widzenia szeregowania zadań istotny jest fakt, iż prezentowany algorytm ASAP umożliwia stosowanie ograniczonej liczby jednostek funkcjonalnych a także operacji, których wykonanie wymaga więcej niż jednego cyklu zegara. Definicje liczby i rodzaju dostępnych jednostek funkcjonalnych oraz liczba cykli zegara (ozn. *CS*) potrzebnych na wykonanie każdej operacji są także (obok grafu *DFG*) parametrami wejściowymi tego algorytmu.

4.1.2. Różnice między opracowanymi algorytmami szeregowania

Operacja v_i (będąca i-tym wierzchołkiem grafu *DFG*) może być przydzielona do jednostki funkcjonalnej fu_j, w kroku (cyklu zegarowym) k (CS_k), jeśli spełnione są poniższe warunki:

- jednostka funkcjonalna fu_j nie ma przydzielonej żadnej innej operacji w kroku k,
- typ jednostki funkcjonalnej fu_j jest zgodny z typem operacji v_i,
- wartości wszystkich danych wejściowych operacji v_i są dostępne najpóźniej z początkiem kroku k.

Jeśli w kroku k więcej niż jedna operacja spełnia powyższe warunki, jako pierwsza do jednostki funkcjonalnej przydzielana jest ta, która posiada najwyższą wartość etykiety p. Etykieta p_i operacji v_i jest równa liczbie kroków potrzebnych na wykonanie wszystkich operacji położonych na ścieżce łączącej operację p_i z najbardziej odległym wyjściem systemowym. Zatem operacje reprezentowane w postaci wierzchołków końcowych posiadają etykietę p równą zero.

Oznacza to, że do jednostek będą przydzielone tylko te operacje, dla których znaleziona zostanie wolna jednostka funkcjonalna. Wszystkie operacje, spełniające drugi i trzeci warunek, które nie zostaną przydzielone w kroku k, będą ponownie wzięte pod uwagę w kroku k+1. Algorytm zakończy pracę, gdy wszystkie jednostki operacje zostaną przydzielone do odpowiednich jednostek funkcjonalnych.

Główna różnica pomiędzy prezentowanymi algorytmami polega na obsłudze sytuacji, gdy dla gotowej do przydzielenia operacji dostępna jest więcej niż jedna jednostka funkcjonalna. Dla tego przypadku algorytm IIOI dokona najprostszego wyboru – przydzieli operację do pierwszej dostępnej jednostki funkcjonalnej. W rezultacie ze wszystkich jednostek funkcjonalnych tego samego typu najbardziej obciążone będą te, o najniższych numerach, gdyż będą zawsze wybierane w pierwszej kolejności. W pewnych sytuacjach, może to wpływać niekorzystnie na dalsze spowalniania takich jednostek, zatem pozostałe dwa algorytmy dokonują bardziej złożonego wyboru jednostki funkcjonalnej spośród zbioru dostępnych.

Algorytm UNILO, dla każdej dostępnej jednostki funkcjonalnej tego samego typu wyznacza wskaźnik obciążenia, który odpowiada liczbie operacji przydzielonych do jednostki funkcjonalnej w krokach poprzednich. Wybierana jest ta jednostka funkcjonalna, która posiada najniższą wartość wskaźnika obciążenia. Powoduje to, w pewnych okolicznościach, korzystne zwiększenie redukcji poboru mocy.

Trzeci z prezentowanych algorytmów (MAREL), wybiera tę jednostkę funkcjonalną, której operacje są najbardziej powiązane z aktualnie przydzielaną operacją v_i. Stopień powiązania jest wyznaczany jako średnia odległości w grafie *DFG* pomiędzy operacją v_i a wszystkimi operacjami przydzielonymi do jednostki funkcjonalnej w poprzednich krokach.

4.2. Spowalnianie jednostek funkcjonalnych

Druga faza przebiega identycznie dla wszystkich trzech prezentowanych algorytmów. Rozpoczyna się ona od umieszczenia wszystkich jednostek funkcjonalnych w zbiorze C_{su}, który reprezentuje wszystkie jednostki funkcjonalne, rozpatrywane celem opóźnienia. Następnie poniższe etapy wykonywane są cyklicznie aż do chwili, gdy zbiór C_{su} będzie pusty.

1. wybór jednostki funkcjonalnej fu_{ms} ze zbioru C_{su},
2. próba spowolnienia jednostki funkcjonalnej fu_{ms},
3. usunięcie jednostki funkcjonalnej fu_{ms} ze zbioru C_{su} w przypadku, gdy jej spowolnienie nie zakończyło się powodzeniem.

Drugą fazę algorytmu szczegółowo przedstawiono w postaci pseudokodu na rys. 3.

```
1  Csu ← SG
2  WHILE ( |Csu| > 0 )
3  {
4   FOREACH ( Ck ∈ Csu )
5    IF ( NOT fsc_fulfiled( Ck ) )
```

```
6      Csu ← Csu \ Ck
7    IF ( |Csu| == 0 )
8     GOTO WHILE_END
9    Cms ← Ck ∈ Csu: fck( Ck )=max
10   IF ( there_are_same_fu( Cms ) )
11    FOREACH ( Ck ∈ Csu, fu(Ck) == fu(Cms) )
12    {
13     C1←next_column_of_the_same_fu(Ck)
14     FOREACH ( vi ∈ Ck, vj ∈ C1 )
15      IF (fvi(vj)>fvi( vi ) )
16       interchange( vi , vj )
17    }
18   SGbackup ← SG
19   FOREACH ( vi ∈ Cms )
20   {
21    insert_idle_operations(vi)
22    IF(NOT delay_all_successors(vi))
23    {
24     SG ← SGbackup
25     Csu ← Csu \ Cms
26     GOTO WHILE_END
27    }
28   }
29   :WHILE_END
30 }
```

Rys. 3. Druga faza algorytmu IIOI

Druga faza algorytmu dokonuje takich zamian operacji pomiędzy kolumnami, aby zysk mocy w ten sposób osiągnięty był maksymalnie duży. Z punktu widzenia wydajności istotne jest także wprowadzenie dodatkowych miar (w postaci funkcji *fsc* i *fck*, opisanych poniżej), pozwalających na szybkie odrzucenie jednostek funkcjonalnych, które nie mogą być spowolnione.

`fsc_fulfiled( Ck )` (wiersz 5)

Funkcja ta dokonuje sprawdzenia warunku wolnego miejsca (ang. *fsc – free space condition*) dla *k*-tej kolumny reprezentującej jednostkę funkcjonalną fu_k. Warunek ten, zdefiniowany poniżej pozwala określić, czy dana kolumna może zostać spowolniona dwukrotnie, bez zwiększenia liczby cykli potrzebnych na wykonanie algorytmu opisanego przez *DFG*.

$$n_i \cdot l_k \le r_o \tag{4.1}$$

gdzie:

n_i – jest liczbą cykli potrzebnych na wykonanie jednej operacji przez jednostkę fu_k,

l_k – jest liczbą wszystkich operacji przyporządkowanych do jednostki fu_k,

r_o – jest liczbą wolnych cykli jednostki fu_k, występujących poniżej pierwszego zajętego cyklu tej jednostki.

Każda kolumna, brana pod uwagę w procesie spowalniania musi spełniać ten warunek. Pomimo faktu, iż nie uwzględnia on zależności pomiędzy operacjami umieszczonymi w innych kolumnach (które powinny być opóźnione w przypadku spowolnienia jed-

nostki fu_k), warunek ten pozwala na szybkie odrzucenie niektórych jednostek na samym początku pętli algorytmu.

fck(C_k) (wiersz 9)

Funkcja ta zwraca całkowitą, znormalizowaną moc potrzebną do wykonania wszystkich operacji w kolumnie C_k (przez jednostkę funkcjonalną fu_k). Jej wartość jest wyrażona wzorem:

$$f_{Ck} = P^n_{dC_k} \cdot l_{Ck} \tag{4.2}$$

gdzie:

P^n_{dCk} – jest znormalizowaną mocą potrzebną na wykonanie jednej operacji realizowanej przez jednostkę funkcjonalną fu_k (przyporządkowaną do kolumny C_k; moc ta jest znormalizowana względem jednostki funkcjonalnej, dla której wartość P_{di} jest najmniejsza),

l_{Ck} – jest liczbą wszystkich operacji przyporządkowanych do kolumny C_k.

Wartość tej funkcji pozwala na rozpoczęcie spowalniania od kolumn, których spowolnienie może przynieść największy zysk, ponieważ jako pierwsze sprawdzane są te kolumny, które posiadają największe wartości wskaźnika *fck*.

5. WYNIKI EKSPERYMENTALNE

Dla opracowanych trzech algorytmów (IIOI, MAREL i UNILO) redukcji mocy pobieranej ze źródła zasilającego przez układy cyfrowe przeprowadzono wiele obliczeń testowych między innymi dla zbiorów benchmarków ISCAS [6]. Wybrane wyniki zestawiono poniżej (CS_{MAX} – maksymalna dopuszczalna liczba cykli, ΔCS – wydłużenie CS_{MAX}).

Tablica 5.1

Redukcje poboru mocy uzyskane dla wybranych przykładów testowych przy pomocy algorytmów IIOI (I), MAREL (M) i UNILO (U)

Przykłady		Zwiększenie CS_{MAX} (ΔCS)								
nazwa układu	liczba bramek	0%			20%			50%		
		I	M	U	I	M	U	I	M	U
s208	104	14.5	14.5	14.5	22.0	31.8	34.2	31.8	46.5	42.9
c1355	514	4.1	3.5	7.7	12.6	13.1	13.1	13.1	13.2	13.2
s5378	2779	8.2	6.6	15.3	17.0	44.8	51.8	41.0	47.1	66.8
s9234	5597	6.4	6.7	3.6	33.0	32.3	65.4	60.7	66.3	67.6

6. PODSUMOWANIE

Zaproponowane podejście minimalizacji mocy pobieranej ze źródła zasilającego P_b poprzez wykorzystanie specjalnie opracowanych technik szeregowania zadań nie prowadzi do zwiększenia czasu opóźnienia całego układu, a jedynie dopuszcza wydłużony okres wykonywania operacji na wybranych jednostkach funkcjonalnych. Należy podkreślić, że szybkość działania jest głównym kryterium przy projektowaniu współczesnych systemów cyfrowych. Wobec powyższego w klasycznym projektowaniu zasadniczy nacisk jest położony na zrealizowanie wymaganych założeń czasowych, po spełnieniu których mogą dopiero być uwzględniane dodatkowe funkcje celu (kosztów), a jedną z nich jest minimalizacja mocy P_b. Z drugiej jednak strony dla większości układów zwiększenie ich szybkości działania pociąga za sobą wzrost poziomu mocy pobieranej ze źródła.

Programy IIOI, MAREL i UNILO wejdą w skład tworzonego systemu, w którym poza wymienionymi znajdą również miejsce algorytmy szeregowania zadań wykorzystujące podejście ewolucyjne.

BIBLIOGRAFIA

[1] Benini L., Bogliolo A. and De Micheli G.: *A survey of design techniques for system-level dynamic power management.* W: IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 8 3, s. 299–316, June 2000.

[2] Michel P. et al.: *The Synthesis Approach to Digital System Design*, Kluwer Academic Publishers, 1992.

[3] Szcześniak W., Voss B., Theisen M, Becker J. and Glesner M.: *Influence of high-level synthesis on average and peak temperatures of CMOS circuits.* W: Microelectronics Journal, vol. 32, s. 855-862, 2001.

[4] Kozieł S. and Szcześniak W.: *Application of adaptive evolutionary algorithm for low power design of CMOS digital circuits.* W: Proc. ICECS'2002, Dubrovnik, s.685-688.

[5] Njolstad T. and Aas E. J.: *Validation of an accurate and simple delay model and its application to voltage scaling*, W: Proc. ISCAS '98, s. 101 - 104, vol.2, 1998.

[6] Collaborative Benchmarking Laboratory, ftp.cbl.ncsu.edu.

ALGORITHMIC METHODS OF POWER REDUCTION IN DIGITAL CMOS CIRCUITS

Summary

This paper presents three heuristic algorithms of high-level synthesis methods for reducing dynamic power for CMOS digital circuits. The main idea was to perform such a scheduling that the clock frequency of some functional units could be diminished (in practice, it is realised by decreasing the power supply of these units). As a result the total power consumption of the circuit can be reduced without deteriorating the total throughput (measured in control steps). The heuristic algorithms are based on three modifications of the Inserting Idle Operation with Interchanging (IIOI) heuristic algorithm. Comparison of the obtained results on the sets of ISCAS'85 and ISCAS'89 benchmarks show that the power reduction varies between 10 and 40% without deteriorating the latency, and up to 70% with extending latency by 50%.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Marek Wroński

Katedra Systemów Mikroelektronicznych, Politechnika Gdańska

NOWE STRUKTURY KORYGOWANYCH LASEREM REZYSTORÓW WARSTWOWYCH ZAPEWNIAJĄCE POPRAWĘ DOKŁADNOŚCI STROJENIA I STABILNOŚCI

Streszczenie

W artykule przedstawiono i przeanalizowano nowe struktury korygowanych laserem rezystorów warstwowych typu „kapelusza" i o kształcie „U" z cięciem prostym i dodatkowym kontaktem przewodzącym (zworą) w ich górnej części. Wprowadzenie zwory umożliwia uzyskanie istotnej poprawy czułościowych charakterystyk strojenia nie tylko zmniejszając wartość zmiany rezystancji względem długości cięcia, ale także zmieniając charakter krzywej z rosnącego na opadający. Jest to bardzo pożądane ze względu na wzrost dokładności strojenia. Następuje także eliminacja lokalnych przegrzewów – tzw. „gorących punktów" występujących na końcu cięcia. W związku z tym można oczekiwać znacznego wzrostu niezawodności i stabilności rezystorów. W artykule zaprezentowano także nowy, oparty na metodzie kolejnych odwzorowań konforemnych, bardzo szybki i efektywny sposób numerycznego wyznaczania charakterystyk strojenia analizowanych struktur.

1. WSTĘP

Korekcja laserowa rezystorów cienkowarstwowych pozwala producentom układów scalonych z dużą dokładnością kontrolować rezystancję. Jednakże korekcja laserowa jest operacją kosztowną, co znacznie zwiększa całkowite koszty produkcji układów scalonych zarówno ze względu na koszt samej korekcji, jak i konieczny wzrost powierzchni układu scalonego. Istotnym elementem staje się zatem optymalizacja wielkości, kształtu i trajektorii nacinania danego rezystora. Zoptymalizowany kształt rezystora dla zadanego zakresu korekcji i przyjętej rozdzielczości minimalizuje zużycie powierzchni chipa, natomiast optymalna trajektoria cięcia laserowego umożliwiają realizację korekcji laserowej z dużą szybkością [1].

W literaturze można znaleźć wiele artykułów dotyczących korekcji laserowej rezystorów [1-4], ale większość z nich opisuje różne techniki korekcji i metody symulujące sam proces korekcji albo też wpływ korekcji na stabilność [2]. Zaledwie kilka artykułów przedstawia problem projektowania korygowanych rezystorów, a problem optymalizacji jedynie został sformułowany bez przedstawienia rozstrzygających rezultatów [4]. Wymaga to wielu symulacji i dużego doświadczenia projektanta. Oznacza to również, że jedynym sposobem na zaprojektowanie rezystora o zadanej specyfikacji jest przyjęcie wstępnej

geometrii rezystora i trajektorii cięcia, przeprowadzenie nacięcia korekcyjnego, a następnie sprawdzenie, czy wyniki spełniają oczekiwany zakres i rozdzielczość [1]. Lecz takie podejście nie daje nam odpowiedzi na pytanie: *czy kształt rezystora i obrana strategia korekcji są najlepsze ze względu na maksymalną dokładność?* Chcemy ostatecznie uzyskać rezystor o konfiguracji z możliwie dużym zakresem zmian rezystancji oraz małą wrażliwością na zmiany długości cięcia, szczególnie przy dochodzeniu wartości rezystancji do wielkości nominalnej. Ponieważ rozdzielczość korekcji jest odwrotnie proporcjonalna do wrażliwości (która jest pierwszą pochodną charakterystyki korekcyjnej), charakterystyka ta odgrywa bardzo ważną rolę przy obliczaniu rozmiarów rezystora [1].

Aby określić całą charakterystykę korekcyjną musimy wyznaczyć zmiany rezystancji krok po kroku w zależności od głębokości cięcia. Wymaga to wielokrotnego rozwiązywania równania Laplace'a [1] lub analizy dużych sieci rezystywnych [3] dla każdego małego kroku na trajektorii cięcia w przypadku stosowania ogólnie przyjętych metod analizy rezystorów z korekcją. O wiele bardziej stosowne i efektywne jest użycie metody kolejnych odwzorowań konforemnych [5,6], umożliwiającej wyznaczenie całej charakterystyki strojenia w jednym procesie analizy. Metoda ta pozwala także na konstruowanie rozwiązania analitycznego potrzebnego do projektowania i optymalizacji.

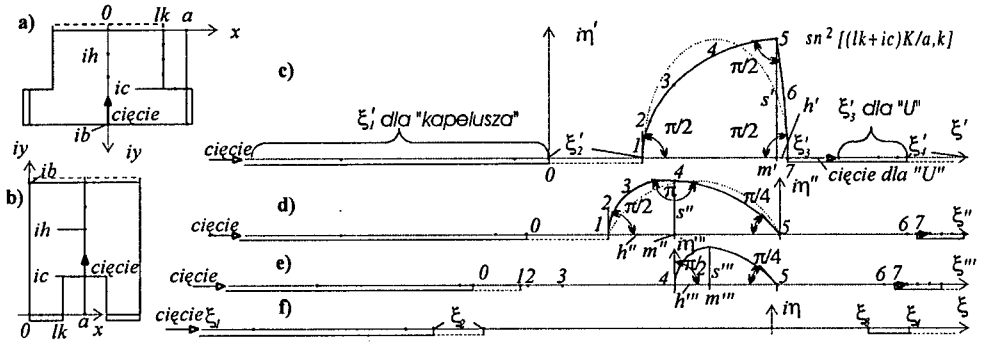
W popularnych rezystorach typu „kapelusza" i w kształcie odwróconej litery „U" z cięciem prostym pośrodku gęstość strumienia prądu nie jest rozłożona równomiernie i może dochodzić do miejscowego przegrzewania oraz efektu wzrostu prądu w obszarach przyległych do cięcia. Ponieważ moc rozpraszana wzrasta z kwadratem gęstości prądu, na zagięciach strumienia prądu (w końcowych punktach cięcia) pojawiają się tzw. „gorące punkty". Ponadto miejscowy gradient napięcia znacznie przekracza ten na znacznej części rezystora. Z tego powodu nie możemy użyć takiej korekcji w przypadku wydzielania znacznych mocy i występowania dużych gradientów napięć. Równomierne wydzielanie ciepła jest pożądane ze względu na wzrost stabilności i efektywniejsze wykorzystanie powierzchni podłoża. Jedną z metod eliminujących miejscowe przegrzewy jest zastosowanie zwór przewodzących wyrównujących rozkład gęstości strumienia prądu [7].

Celem niniejszej pracy jest pokazanie w jaki sposób wprowadzenie zworek (dodatkowych nigdzie nie podłączonych kontaktów przewodzących) w górnej części rezystorów typu „kapelusza" i typu „U" wpływa na zmianę charakterystyk korekcyjnych, a także jak przejść od analizy do projektowania i optymalizacji takich struktur.

2. ANALIZA REZYSTORÓW TYPU „KAPELUSZA" I TYPU „U" ZE ZWORĄ PRZEWODZĄCĄ

Ze względu na symetrię osiową rezystorów typu „kapelusza" i typu „U" wystarczy analizować połówki struktur przedstawionych na rys.1a i 1b. Zworkę przewodzącą zaznaczono przerywaną linią. Analizę takich struktur można przeprowadzić w sposób analogiczny do przedstawionego w pracy [8]. Dla przykładu wzięliśmy rezystor typu „kapelusza" analizowany inną metodą w [1] oraz rezystor w kształcie litery „U" mający taką samą początkową (bez cięcia korekcyjnego) wartość rezystancji, syntezę którego przeprowadzono w punkcie 3.2.

Po odwzorowaniu połówki struktury na górną półpłaszczyznę przy użyciu funkcji eliptycznej $sn^2(2Kz/D, k)$, gdzie K jest całką eliptyczną zupełną pierwszego rodzaju o module k określonym równaniem $D/L=2K(k)/K(k')$, otrzymujemy obszar w postaci półpłaszczyzny z wyciętym segmentem w postaci trójkąta krzywoliniowego pokazany na rys.1c.



Rys.1. Analizowane konfiguracje korygowanych laserem rezystorów warstwowych typu „kapelusza" (a) i typu „U" (b) oraz początkowe kroki odwzorowania konforemnego połowy struktury (pierwszej ćwiartki) z rys.1a na górną półpłaszczyznę (c, d, e, f).

Z powodu kątów prostych w wierzchołkach trudno jest znaleźć elementarne funkcje odwzorowujące ten obszar na półpłaszczyznę, takie żeby uzyskać dużą dokładność odwzorowania. Dlatego też opracowaliśmy nową metodę odwzorowania konforemnego podobnych obszarów na półpłaszczyznę, która jest najbardziej efektywna dla dokładnych obliczeń komputerowych.

Metoda ta wykorzystuje wielokrotnie tylko jedną elementarną funkcję odwzorowującą *Ehs(m, h, s)* [5], która wyprowadza wszystkie punkty półokręgu na oś rzeczywistą, jednak nie w celu eliminacji przecięć jak w [5], lecz dla eliminacji części wycięcia wyprowadzając początkową lub końcową część brzegu na oś rzeczywistą. Łuk 5∩7 na rys.1c jest dobrze aproksymowany przez półokrąg, można go więc odwzorować z dużą dokładnością na oś rzeczywistą. W rezultacie otrzymujemy mniejsze wycięcie o podobnym kształcie, również z kątem prostym na początku. Wynika to z właściwości funkcji odwzorowującej *Ehs(m, h, s)*, która powoduje dwukrotne zmniejszenie kąta pomiędzy łukiem (który jest styczny) a wklęsłością w wierzchołku 4 oraz dwukrotne zwiększenie kąta przy podstawie. Po kilku takich krokach możemy uzyskać czystą półpłaszczyznę (bez wklęsłości i wybrzuszeń) pokazaną na rys.1f, zaś rezystancję rezystorów bez dodatkowej zworki i razem z nią (pokazaną na rys.1 linią przerywaną) możemy łatwo obliczyć na podstawie otrzymanych współrzędnych punktów końcowych kontaktów na półpłaszczyźnieξ_i [6]:

$$R = R_0 \frac{K(k')}{K(k)} = -R_0 \frac{\pi}{\ln q}, \qquad k^2 = 1 - k'^2 = \frac{(\xi_4 - \xi_1)(\xi_3 - \xi_2)}{(\xi_4 - \xi_2)(\xi_3 - \xi_1)} \tag{2.1}$$

gdzie: $x = 0.5 \cdot \left(1 - \sqrt{k'}\right) \Big/ \left(1 + \sqrt{k'}\right)$; $\qquad q = x + 2x^5 + 15x^9 + 150x^{13} + 1707x^{17} + \ldots$ (2.2)

K(k) – całka eliptyczna zupełna pierwszego rodzaju,

R_O – rezystancja na kwadrat warstwy rezystywnej.

Jeżeli w opisanym wyżej procesie eliminowania wycięcia w półpłaszczyźnie będą odwzorowywane także współrzędne punktów osi urojonej, które odpowiadają końcowym punktom cięcia korekcyjnego, to możemy otrzymać całą charakterystykę korekcyjną.

2.1. Uzyskiwanie analitycznego wzoru dla funkcji odwzorowującej na półpłaszczyznę

Ze względu na zjawisko nierównomiernego rozpływu prądu w rezystorach typu „kapelusza" i w kształcie odwróconej litery „U" obliczenie efektywnej liczby kwadratów nie odbywa się w sposób bezpośredni. Do projektowania i optymalizacji takich struktur pożądane są względnie proste, lecz stosunkowo dokładne wyrażenia analityczne odwzorowania konforemnego na półpłaszczyznę. Najprostszy wzór analityczny można uzyskać przez aproksymację wklęsłości w półpłaszczyźnie przy pomocy półelipsy (linia kropkowana na rys.1c) i jednokrotne użycie elementarnej funkcji *En(m, a, b)* [5], która odwzorowuje półpłaszczyznę z wyciętą połową elipsy w całkowitą półpłaszczyznę. Parametry a – poziomą półoś elipsy oraz m – współrzędna środka elipsy, można uzyskać według współrzędnych wierzchołków elipsy przy podstawie, natomiast ostatni parametr b – pionowa półoś może być wzięty równy maksymalnej wartości urojonej (w trzecim wierzchołku). Takie podejście zapewnia nam dokładność rzędu 1%, wystarczającą do obliczenia początkowej wartości rezystancji.

Bardziej dokładny wzór zapewniający dokładność rzędu ułamka procenta można skonstruować stosując dwa kroki wyprowadzania na oś rzeczywistą punktów wycięcia w półpłaszczyźnie. Można to osiągnąć na przykład stosując najpierw elementarną funkcję *Ehs(m, h, s)*, a następnie funkcję *En(m, a, b)* lub *Ln(m, a, β)* [5] (ta ostatnia funkcja odwzorowuje półpłaszczyznę z wyciętym kolistym segmentem na całą półpłaszczyznę). W tym przypadku aby określić parametry musimy ponownie obliczyć pozycje końców kontaktów oraz współrzędną jakiegoś dodatkowego punktu, np. punktu o maksymalnej wartości urojonej.

3. SYNTEZA I OPTYMALIZACJA REZYSTORÓW TYPU „KAPELUSZA" I TYPU „U"

Do projektowania konieczne jest rozwiązanie zagadnienia odwrotnego, tj. znalezienie parametrów geometrycznych rezystorów na przykład na podstawie zadanej początkowej liczby kwadratów R_{IN}/R_O bez cięcia korekcyjnego lub z cięciem o zadanej długości h. Wychodząc z tej wartości możemy znaleźć odpowiednie rozmieszczenie kontaktów na górnej półpłaszczyźnie. Wówczas kolejnym krokiem jest określenie na podstawie odwrotnego odwzorowania konforemnego odpowiedniego kształtu geometrycznego.

3.1. Synteza na półpłaszczyźnie

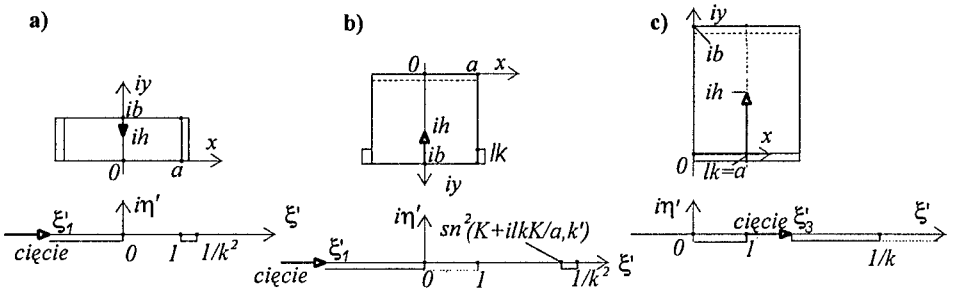
Ponieważ przekształcenia konforemne są jednoznacznie określone, z wyjątkiem trzech dowolnych punktów (np. współrzędne końcowych punktów kontaktów), pierwszy problem sprowadza się do rozwiązania jednego równania nieliniowego (2.1) z całkami eliptycznymi względem modułu k. Przyjmując normalizację $\{\xi_1, \xi_2, \xi_3\}=\{-\infty, 0, 1\}$, ostatnia współrzędna równa się $\xi_4=1/k^2$. Moduł k można łatwo obliczyć używając szybko zbieżnego rozwinięcia w szereg (2.2), podstawiając do (2.1) wartość R_O/R_{IN} zamiast $K(k')/K(k)$.

3.2. Odwrotne odwzorowanie konforemne i optymalizacja kształtu struktury

Aby rozwiązać drugi problem musimy znaleźć odwrotną funkcję odwzorowującą. Jest ona odwrotnością kwadratu funkcji eliptycznej $sn^2(\xi K/a, k)$, jeżeli chcemy, żeby ostateczny wynik był przetransponowany z półpłaszczyzny na jeden z obszarów przedstawionych na rys.2. Te różne struktury uzyskano zmieniając przyjmowaną a priori normalizację $\{\xi_1, \xi_2, \xi_4\}$ w inną $\{\xi'_1, \xi'_2, \xi'_4\}$ przy pomocy biliniowej transformacji konforemnej (3.1), która odwzorowuje półpłaszczyznę ζ w inną półpłaszczyznę ζ'.

$$((\zeta'-\xi'_1)/(\zeta'-\xi'_2))\cdot((\xi'_4-\xi'_2)/(\xi'_4-\xi'_1))=((\zeta-\xi_1)/(\zeta-\xi_2))\cdot((\xi_4-\xi_2)/(\xi_4-\xi_1)) \quad (3.1)$$

Z (3.1) uzyskujemy inną współrzędną kontaktu $\xi'_3=\zeta'(\xi_3)$, odpowiedzialną za nową podaną normalizację. Jeśli dana jest długość cięcia h, musimy wziąć $\xi'_1=sn^2(iKh/a, k)$ dla struktury na rys.2b (oraz $\xi'_3=sn^2(K+iKh/a, k)$ dla rys.2c). Dla rezystora z rys.2b założyliśmy wymiary: h=0, a=95 μm, b=160 μm, więc k=0.29263, i lk=19.8 μm (k=0.29101, i lk=18.6 μm dla rezystora ze zworą). Struktura przedstawiona na rys.3c została zaprojektowana tak, by uzyskać pierwotne cięcie o głębokości 50%. W tym przypadku $\zeta'=\xi'_3=1/k$ i $\xi'_4=1/k^2$. Zatem z (3.1) wynika, że k=0.093820, co oznacza, że $b=2.39a$ (oraz k=0.095911, $b=2.37a$ dla rezystora ze zworą).



Rys.2. Równoważne struktury prostokątnych rezystorów warstwowych posiadające taką samą wartość początkowej rezystancji: a) zwykły rezystor prostokątny z kontaktami na przeciwległych bokach, b) analogiczny rezystor prostokątny z krótszymi kontaktami, c) rezystor prostokątny z kontaktami na jednym boku oddzielonymi cięciem.

Jeżeli potrzebujemy przenieść rozwiązanie z półpłaszczyzny na obszar rezystora typu „kapelusza" lub „U" z wyciętym obszarem prostokątnym, musimy wstawić do (3.1) nie tylko współrzędne, ale również wyrażenie analityczne odpowiedzialne za te współrzędne. Na przykład, używając jednokrokowego wzoru analitycznego uzyskanego według opisu w punkcie 2.2 musimy użyć wyrażeń:

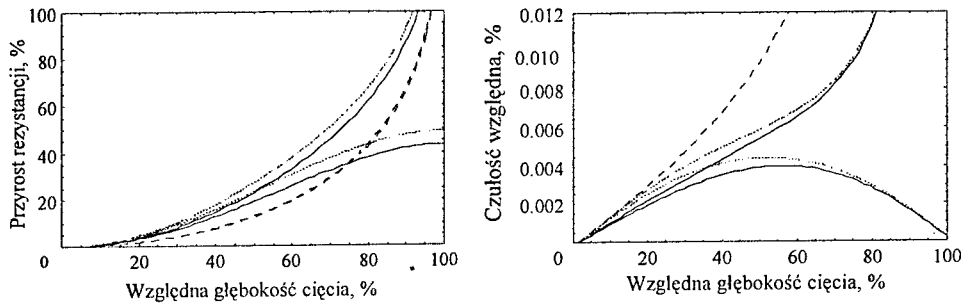
$$\begin{aligned} \xi'_1 &= \frac{-m\cdot a+b\cdot\sqrt{m^2+b^2-a^2}}{a-b}; & \xi'_4 &= \frac{(-m+1/k^2)\cdot a+b\cdot\sqrt{(-m+1/k^2)^2+b^2-a^2}}{a-b} \\ \xi'_2 &= m-a-b; & \xi'_3 &= m+a+b; \end{aligned} \quad (3.2)$$

jako współrzędnych kontaktów z przybliżonymi wartościami parametrów a i b znanymi z analizy (lub z uzyskanych początkowych wartości współrzędnych). Wówczas musimy

rozwiązać (3.1) dla parametru m (współrzędnej środka wycięcia), obliczyć $\xi'_3=m+a+b$ i wreszcie użyć funkcji odwrotnej do funkcji $sn^2(\xi'_3K/a,\ k)$. Używając tego rozwiązania jako wartości początkowej oraz dwukrokowego wzoru analitycznego, wyprowadzenie którego opisano krótko w punkcie 2.2, skonstruowaliśmy pewien proces iteracyjny i znaleźliśmy dokładne rozwiązanie $lk=0.75a$, $c=0.284b$, przedstawione na rys.1b (a także dla $lk=0.5a$ uzyskaliśmy rozwiązanie wynoszące $c=0.121b$).

4. WYNIKI PRZEPROWADZONYCH SYMULACJI KOMPUTEROWYCH

Wyniki obliczeń przeprowadzonych według metody opisanej w rozdziale 2 zaprezentowano na rysunkach 3 i 4. Przedstawiono na nich charakterystyki korekcyjne, czyli przyrostu rezystancji względem wartości początkowych (tj. bez cięcia korekcyjnego) oraz charakterystyki czułościowe, czyli względnego przyrostu rezystancji odniesionego do przyrostu głębokości cięcia, względem zmiany głębokości cięcia (odniesionego do wysokości rezystora). Charakterystyki czułościowe uzyskano za pomocą numerycznego różniczkowania charakterystyk korekcyjnych.



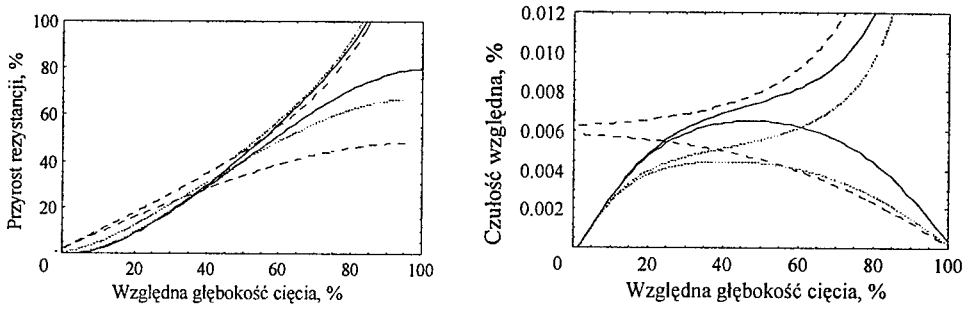
Rys.3. Charakterystyki korekcyjne i czułościowe dla rezystora typu „kapelusza" (linia ciągła) oraz rezystorów prostokątnych z rys.2a (linia przerywana) i z rys.2b (linia kropkowana).

W oparciu o uzyskane charakterystyki możemy wyciągnąć kilka wniosków. Przede wszystkim należy zaznaczyć, że uzyskane charakterystyki korekcyjne dla struktury z rys.1a są takie same jak w [1], ale czas obliczeń był krótszy o co najmniej dwa rzędy wielkości. Potwierdza to skuteczność opracowanej metody.

Zworka przewodząca przy górnej krawędzi rezystora tylko nieznacznie zmniejsza wartość początkową rezystancji (bez nacięcia korekcyjnego), tj. nie tak istotnie jak w przypadku umieszczenia dodatkowego kontaktu przewodzącego pośrodku struktury, jak to miało miejsce w przypadku struktur analizowanych w pracy [9]. Jednak teraz cięcie można poprowadzić aż do granicy tej zworki, tak iż zakres trymowania także ulega zwiększeniu. Ponadto zworka ta w sposób istotny wpływa na charakterystyki czułościowe nie tylko zmniejszając absolutne wartość czułości, ale także zmieniając kształt krzywej z rosnącej na malejącą, jeżeli cięcie jest dłuższe niż połowa wysokości rezystora. Jest to wysoce pożądane dla osiągnięcia lepszej rozdzielczości w procesie korygowania.

Bez zwory przewodzącej zakres korekcji nie może przekroczyć pewnej górnej granicy rezystancji (np. 40% zmiany rezystancji), a cięcie nie powinno być dłuższe niż 50% wysokości (albo powinno się kończyć przynajmniej na długość kontaktu poniżej górnej krawędzi rezystora). Cięcie to zawsze kończy się przy maksymalnej wrażliwości, dlatego

też dla korekcji o wysokiej dokładności stosowane jest dodatkowo drugie precyzyjne cięcie w celu osiągnięcia końcowej żądanej wartości rezystancji i zadanej rozdzielczości. Zazwyczaj przeprowadza się je równolegle do cięcia pierwszego. Z powodu zatrzymywania i cofania korektora laserowego i ponownego jego uruchamiania komplikuje to znacznie proces i zwiększa koszty korekcji. Dodatkowa zwora nie tylko eliminuje lokalne przegrzewy (tzw. „gorące punkty"), związane ze zwiększeniem gęstości prądu na końcu cięcia, ale również polepsza warunki odprowadzenia ciepła. W związku z tym osiągana maksymalna temperatura będzie niższa, co powoduje, że rezystory ze zworką przewodzącą będą starzeć się wolniej.



Rys.4. Charakterystyki korekcyjne i czułościowe dla rezystora typu „U" o długości kontaktów lk=0.5a (linia ciągła) oraz długości lk=0.75a (linia kropkowana), a także rezystora prostokątnego z rys.2c (linia przerywana).

Na podstawie rys.4 można zauważyć, że zmiany normalizacji i wielkości wyciętego odcinka mają istotny wpływ na charakterystyki wrażliwościowe. Powinny zatem być one brane pod uwagę przy wykonywaniu procesu optymalizacji w oparciu o wybrane kryterium, takie jak maksymalna wrażliwość korekcji w rozpatrywanych obszarach.

5. WNIOSKI

Prawidłowo rozpatrzone konfiguracje korygowanych rezystorów pozwalają uzyskać znacznie bardziej jednorodny rozkład mocy i w konsekwencji zredukować do minimum wymiary rezystora. Zaproponowane tu konfiguracje i metody ich analizowania, syntezy i optymalizacji umożliwiają wytwórcom układów scalonych wybór pożądanego rezystora do korekcji przy minimalnym nakładzie kosztów na badania. Struktury takie pozwalają osiągnąć wyższą stabilność rezystorów i większą ich dokładność. Opracowane w artykule uniwersalne metody są bardzo pomocnym dodatkowym narzędziem stosowanym w projektowaniu komputerowym, które łatwo można zintegrować z istniejącym oprogramowaniem CAD. Otrzymane charakterystyki korekcyjne mogą być użyte bezpośrednio do kontroli procesu korekcji w czasie rzeczywistym.

BIBLIOGRAFIA

[1] Schimmanz K., Jacobsen S.M.: *Film Resistor Design for High Precision Laser Trimming.* German preprint BTU-Cottbus M-02/02, 13 pages.
[2] Bube K.R., Miller A.Z., Howe A., Antoni B.: *Influence of Laser-Trim Configuration on Stability of Small Thick-Film Resistors,* W: Solid State Technology, Nov.1978, pp.55-60.
[3]Ronniger R., *Experience with Trimming of Chip Resistors and how to Improve Trimming Results with CAD Simulation and Thick Film Process Control,* W: Hybrid Circuits, No.9, Jan.1986, pp.40-43.
[4] Chaudhry M.A., *Design Optimisation of Integrated Circuit Thin-film Resistors with Slit and Curved Boundaries,* W: Microelectronics International, No.40, May 1996, pp.17-21.
[5] Filczakov P.F, Pribliżennyje Metody Konformnych Otobrażenij, N.Dumka, Kijev, 1964.
[6] Wroński M., Analysis, Synthesis and Optimisation of Multicontact Distributed Resistive Networks, W: Proc.of the 12th ECCTD, vol.2, Istanbul, Turkey, Aug.1995, pp.925-928.
[7] Jermołajev Ju.P., Perspektivnyje I racjonalnyje konstrukcji pieczatnych soprotivlenij w mikrominjaturnoj apparature, W: Izvestja Vuzov ZSRR – Raditechnika, t.5, No. 4, 1962, str. 469-475.
[8] Chawala B. R. and Gummel H. K., A boundary technique for calculation of distributed resistanc, W: IEEE Trans. on Electron Devices, vol. ED-17, Oct. 1970, pp. 915-925.
[9] Wroński M., Sposób korekcji wartości rezystancji rezystorów warstwowych, Zgłoszenie patentowe P.356950 z dn. 4.11.2002, 11str.

NEW LASER TRIM FILM RESISTORS WITH IMPROVED RESOLUTION AND STABILITY

Summary

In this paper new laser trim structures of the popular top-hat and U-shaped resistors with a plunge cut and with an additional contact (short-circuit jumper) on the top of such resistors are presented and analysed. This enables us to achieve substantial improvement of sensitivity characteristics not only by decreasing the value of the rate of resistance change with respect to the trim depth, but also by changing the shape of the curve from rising to falling down, which is most desired in the trimming process as higher resolution can be achieved. Also local overheating (so-called "hot zone") at the end of the cut is eliminated, so higher reliability and stability can be achieved. Basing on the method of successive conformal mappings, a new effective method for computing trim characteristics of such structures has been elaborated as well.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Marek Wroński

Katedra Systemów Mikroelektronicznych, Politechnika Gdańska

NOWE PODEJŚCIE DO PROJEKTOWANIA I REALIZACJI KOREKCJI LASEROWEJ REZYSTORÓW WARSTWOWYCH

Streszczenie

W artykule przedstawiono i przeanalizowano nowe podejście do projektowania i realizacji rezystorów warstwowych korygowanych laserem, polegające na wprowadzaniu dodatkowego kontaktu przewodzącego, wokół którego realizuje się następnie nacięcie korekcyjne. Powoduje to zmniejszenie rezystancji pomiędzy pozostałymi dwoma kontaktami, tj. rezystancji początkowej, co oznacza uproszczenie projektowania korygowanych rezystorów, a także zwiększenie dopuszczalnego zakresu zmian rezystancji. Następuje także poprawa czułościowych charakterystyk strojenia, a ponieważ osiągnięcie nominalnej wartości rezystancji następuje przy krótszej długości cięcia (lub mniejszych rozmiarach rezystorów), to ma to wpływ na zwiększenie szybkości procesu korekcji i zmniejszenie ceny produkowanych układów scalonych. Dodatkowy kontakt polepsza także odprowadzenia ciepła wydzielanego w rezystorze co z kolei wpływa na wzrost stabilności.

1. WSTĘP

Funkcjonalność, możliwości i niezawodność współczesnych hybrydowych układów scalonych zależy od dokładnych wartości rezystancji lub ich stosunków. W praktyce jednakże precyzyjne rezystory bardzo trudno zrealizować. Wynika to z naturalnego rozrzutu i dryftu czasowego rezystorów cienkowarstwowych, jak również ze zmian procesu ich wytwarzania. Laserowa korekcja funkcjonalna stała się najbardziej popularną metodą dostosowywania każdego układu hybrydowego do wymagań precyzyjnych rezystorów [1]. Jest wysoce pożądane, aby optymalizować wielkość, kształt i długość nacięcia rezystora, który ma być korygowany, ze względu na wysokie koszty związane z realizacją procesu korekcji [2-4]. Dla przykładu, optymalny kształt minimalizuje zużytą powierzchnię chipu, a zoptymalizowanie trajektorii nacięcia skutkuje korekcją niezawodną i realizowaną w sposób całkowicie automatyczny z bardzo dużą szybkością. Jednak ciągle aktualnym pozostaje pytanie, *jaki kształt rezystora i jaka trajektoria jest najbardziej optymalna*, tzn. zapewniająca najkrótsze cięcie (co oznacza maksymalną prędkość procesu korekcji), największą dokładność i wystarczający dla pokrycia rozrzutów technologicznych zakres zmian wartości rezystancji. Proces optymalizacji rozpoczyna się na etapie projektu, a kończy na etapie wielu prób z samym procesem korekcji. Funkcją matematyczną, znajomość której

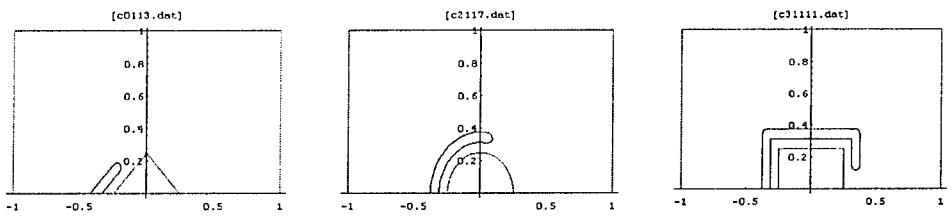
jest konieczna do optymalizacji, jest tak zwana charakterystyka korekcji. Opisuje ona zmianę rezystancji w funkcji długości trajektorii cięcia.

W artykule opisano nowe struktury korygowanych rezystorów warstwowych, metody ich analizy i wyniki przeprowadzonych symulacji komputerowych oraz wnioski co do optymalnej strategii prowadzenia procesu korekcji.

2. WYKORZYSTANIE DO KOREKCJI STRUKTUR REZYSTYWNYCH Z DODATKOWYM KONTAKTEM PRZEWODZĄCYM

Znane dotychczas sposoby korekcji laserowej rezystorów warstwowych polegały na dokonaniu wcięcia w warstwie rezystywnej rezystora dwu-kontaktowego. Wielkość wcięcia jest ustalana i kontrolowana w czasie procesu korekcji, tak aby uzyskać zadaną (nominalną) wartość rezystancji. Z uwagi na to, że wcięcie powoduje zawsze wzrost rezystancji, rezystor podlegający korekcji powinien być zaprojektowany w taki sposób, aby posiadał wartość rezystancji początkowej mniejszą od nominalnej z uwzględnieniem możliwych rozrzutów technologicznych.

W [5] zaproponowaliśmy nowe podejście polegające na wykonywaniu w rezystorze dodatkowego trzeciego kontaktu przewodzącego, wokół którego będzie następnie realizowane cięcie korekcyjne (patrz rys.1). Wprowadzenie dodatkowego kontaktu powoduje zawsze zmniejszenie rezystancji pomiędzy pozostałymi dwoma kontaktami, tj. rezystancji początkowej (bez nacięcia korekcyjnego), dzięki czemu można uniknąć projektowania rezystora na mniejszą wartość rezystancji. To także, przy niezmienionej górnej wartości rezystancji, określonej przez całkowite odcięcie tegoż kontaktu, oznacza zwiększenie dopuszczalnego zakresu zmian rezystancji korygowanych rezystorów. Następuje również poprawa charakterystyk czułościowych, a ponieważ przy zachowaniu dopuszczalnego zakresu korekcji (zmian rezystancji) i przy tych samych rozrzutach technologicznych osiągnięcie nominalnej wartości rezystancji następuje przy krótszej długości cięcia (lub mniejszych jego rozmiarach), to wpływa to z kolei na zwiększenie szybkości procesu korekcji i zmniejszenie ceny produkowanych układów scalonych. Dodatkowy kontakt polepsza także odprowadzenia ciepła wydzielanego w rezystorze co skutkuje zmniejszeniem maksymalnej temperatury i wzrostem jego stabilności. Może on być dodatkowo wykorzystany do bardziej precyzyjnego (trzypunktowego) pomiaru wartości rezystancji przez co osiąga się poprawę kontroli procesu korekcji. Chyba jedyną wadą takiego podejścia jest nieznaczny wzrost szumów związanych z istnieniem dodatkowego kontaktu, jednak jest on na ogół mały, gdyż tylko niewielka część prądu jest zamykana poprzez ten częściowo odcięty kontakt.



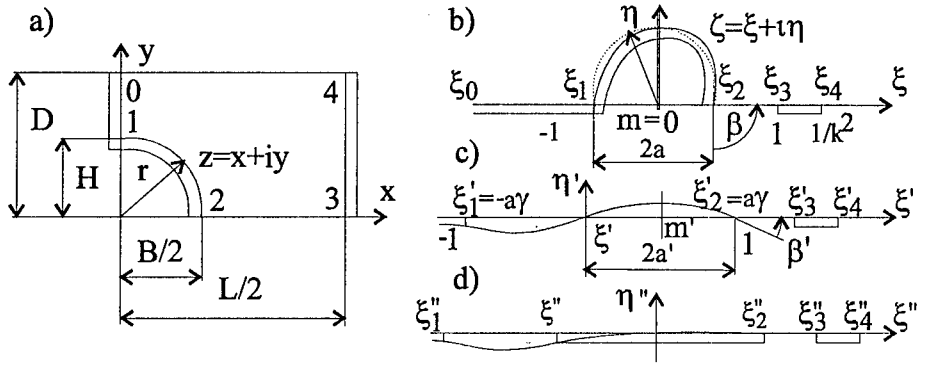
Rys.1. Przykładowe struktury rezystorów prostokątnych z wprowadzonym dodatkowym kontaktem.

3. METODY ANALIZY I SYMULACJI PROCESU KOREKCJI

Aby uzyskać całą charakterystykę korekcyjną należy obliczyć rezystancję dla każdej małej zmiany geometrycznej, którą wywołuje każdy impuls lasera na swej drodze cięcia. Wymaga to pracochłonnego procesu wielokrotnego rozwiązywania równania Laplace'a, np. metodą różnic skończonych. Ponieważ równanie Laplace'a, a więc i wartość rezystancji, jest inwariantne względem odwzorowań konforemnych więc problem analizy może zostać zredukowany do odwzorowania konforemnego zadanego obszaru rezystywnego na inny obszar, np. górną półpłaszczyznę dla której wartość rezystancji określona jest przez współrzędne końcowych punktów kontaktów ξ_i [6]. Zaś samo odwzorowanie konforemne może być zrealizowane numerycznie, np. bardzo efektywną metodą kolejnych odwzorowań konforemnych [7]. Polega ona na odwzorowaniu wyjściowego obszaru rezystywnego na półpłaszczyznę z pewnym krzywoliniowym wycięciem[1] które następnie wyciągane na oś rzeczywistą za pomocą prostych funkcji analitycznych.

3.1. Analiza zakresu korekcji dla cięć przy dodatkowym symetrycznym kontakcie przewodzącym

Po wprowadzeniu dodatkowego kontaktu przewodzącego otrzymujemy strukturę trzy-kontaktową którą można opisać za pomocą macierzy impedancyjnej lub admitancyjnej, bądź określając parametry czwórników typu "Π" lub "T" (oznaczane dalej jako r_i). Stosując twierdzenie Bartletta analizę struktur trzy-kontaktowych z symetrią osiową można sprowadzić do dwukrotnie przeprowadzonej procedury analizy struktury połówkowej ze zwarciem i rozwarciem wzdłuż osi symetrii.



Rys.2. Analizowana połówka rezystora (a) i początkowe kroki jej odwzorowania konforemnego na półpłaszczyznę (b,c,d), gdzie: b) $\zeta=sn^2(2Kz/L,k)$ z $L/D=2K(k)/K(k')$,
c) $\zeta'=L_n(m,a,\beta)=a\gamma[(\zeta-m+a)^\gamma+(\zeta-m+a)^\gamma]/[(\zeta-m+a)^\gamma+(\zeta-m+a)^\gamma]$ z $\gamma=\pi/(\pi-\beta)$,
d) $\zeta''=L_n(m',a',\beta')$.

Ponieważ otrzymane w ten sposób struktury są już dwu-kontaktowe więc stosując opisaną wyżej procedurę eliminowania wklęsłości w półpłaszczyźnie, związanej z istnieniem dodatkowego kontaktu, co można osiągnąć stosując kilkakrotnie (w zależności od

[1] W naszym przypadku funkcją odwzorowującą może być funkcja eliptyczna $\zeta=sn^2(2Kz/L,k)$ z $L/D=2K(k)/K(k')$, zaś wycięcie będzie odpowiadało dodatkowemu kontaktowi przewodzącemu.

wymaganej dokładności) funkcję analityczną $L_n(m,a,\beta)$ [8] można określić współrzędne na osi rzeczywistej końcowych punktów kontaktów ξ_i i na ich podstawie obliczyć wartość rezystancji stosując wzór (3.1).

Zakres korekcji zdefiniowany jest przez dwie wartości rezystancji. Pierwsza z nich jest podwojoną wartością rezystancji pomiędzy dwoma kontaktami zwartej pośrodku struktury połówkowej, z uwzględnieniem połówki dodatkowego kontaktu $R_{12}=2r_1=2r_{short}$ i może być wyznaczona ze wzoru:

$$R_{12} = 2R_0 \frac{K(k)}{K(k')} = \frac{2R_0}{\pi} \ln\left(\frac{1}{q'}\right) \cong \frac{2R_0}{\pi} \ln\left(2 \cdot \frac{1+\sqrt{k}}{1-\sqrt{k}}\right), \qquad (3.1)$$

gdzie $$k^2 = 1-k'^2 = (\xi_3''-\xi_2'')/(\xi_4''-\xi_2'') \qquad (3.2)$$

Druga wartość rezystancji, która odpowiada całkowitemu odcięciu dodatkowego kontaktu (z łukiem 12 na rys.2 jako zwykłym brzegiem nie przewodzącym) $R_{12C}=2r_{1C}$ jest obliczana z użyciem tych samych wzorów (3.1) i (3.2) tylko z ξ_2'' zastąpionym przez ξ_1''.

3.2. Wyznaczanie charakterystyki korekcyjnej dla cięć przy kontakcie

Jeśli trzeci kontakt zostanie odcięty tylko częściowo, nie ma symetrii i wówczas wszystkie parametry r_i schematu zastępczego typy „T" wyrażają się przez współczynniki $a_{k,m}$ w formie całek hipereliptycznych, które jednak łatwo sprowadzić do kwadratury Gaussa-Czebyszewa (3.3) i wyznaczyć numerycznie [6].

$$a_{k,m} = \int_{\xi_k}^{\xi_{k+1}} \xi^m \Big/ \sqrt{\prod_{i=1}^{6} |\xi - \xi_i|} \cdot d\xi \cong \frac{\pi}{j} \sum_{i=1}^{j} f\left(\cos\frac{2i-1}{2j}\pi\right), \qquad (3.3)$$

gdzie:

$$\xi(t) = (\xi_{k+1} - \xi_k)\cdot t/2 + (\xi_{k+1} + \xi_k)/2; \quad f(t) = \xi(t)^m \Big/ \sqrt{\prod_{i=1, i\neq k, k+1}^{6} |\xi(t) - \xi_i|}$$

Wartość zaś rezystancji R_{12} może być otrzymana z rozwiązania układu równań liniowych (7) w [6] i wynosi:

$$R_{12} = r_1 + r_2 = \frac{a_{3,0}(a_{4,1} - a_{2,1}) - a_{3,1}(a_{4,0} - a_{2,0})}{a_{3,1} \cdot a_{1,0} - a_{3,0} \cdot a_{1,1}} \qquad (3.4)$$

Jeśli cięcie ma określoną szerokość proces numerycznego odwzorowania na półpłaszczyznę jest analogiczny z tą jedynie różnicą, że wymaga on dodatkowego kroku realizowanego przy użyciu funkcji $L_n(m,a,\beta)$ [8] dla wyciągania na oś rzeczywistą półpłaszczyzny punktów wgłębienia związanego z grubością cięcia.

$$\zeta = \xi + i\eta = L_n(m,a,\beta) = a\gamma \frac{(z-m+a)^\gamma + (z-m-a)^\gamma}{(z-m+a)^\gamma - (z-m-a)^\gamma}; \qquad \gamma = \frac{\pi}{\pi-\beta}, \qquad (3.5)$$

3.3. Wyznaczanie charakterystyki korekcyjnej dla cięć oddalonych od kontaktu

Jeśli cięcie prowadzone jest w pewnej odległości od dodatkowego kontaktu przewodzącego, wówczas po wyprowadzeniu na oś rzeczywistą punktów tego kontaktu następnie należy wyeliminować cięcie np. stosując kilkakrotnie elementarną funkcję $E_{hs}(m,h,s)$ [8] w celu wyprowadzenia na oś rzeczywistą punktów cięcia.

$$\zeta = \xi + i\eta = E_{hs}(m,h,s) = \frac{\sqrt{(z-m)^2+s^2}}{1+\varepsilon\left(z-m-\sqrt{(z-m)^2+s^2}\right)}\ ; \qquad \varepsilon = \frac{h}{s^2}\ , \tag{3.6}$$

Dla długich cięć wskazane jest, by dzielić je i usuwać kawałek po kawałku. Oddalenie zaś cięcia od kontaktu nieznacznie zmienia położenie punktów końcowych kontaktów na półpłaszczyźnie co wpływa na zmianę rezystancji.

3.4. Uzyskiwanie wyrażeń aproksymujących zależność rezystancji od długości cięcia

Gdy jest jednoznacznie określona kolejność i sposób wyznaczania parametrów odwzorowań elementarnych, łącząc poszczególne odwzorowania w jedno można uzyskać wyrażenie analityczne dla odwzorowania wyjściowego obszaru rezystywnego na półpłaszczyznę. Choć nie jest to konieczne przy obliczeniach rezystancji, ale zależności takie mogą być wykorzystane do obliczeń w czasie rzeczywistym koniecznych przy korekcji algorytmicznej. Dla uzyskania stosunkowo prostych i dość dokładnych wyrażeń analitycznych jest pożądane przy zadanej liczbie kroków odwzorowań elementarnych określenie optymalnych wartości parametrów, takich żeby uzyskać maksymalną dokładność odwzorowania (patrz np. [9,10]).

W przypadku uzyskanego niesymetrycznego rozmieszczenia na półpłaszczyźnie punktów końcowych kontaktów nie można skorzystać z prostszych wzorów (3.1) i (3.2) lecz należy obliczenia prowadzić w oparciu o wzór (3.3) przyjmując określoną liczbę węzłów kwadratury. Stwierdziliśmy, że przyjmując j=3 uzyskuje się błąd obliczenia R_{12} ok. 0.1% (zaś dla j = 2 błąd ok. 1%).

W przypadku cięć przy kontakcie można także uzyskać bardziej proste wyrażenia analityczne (z dokładnością rzędu kilku %) stosując proces symetryzacji podcinanego kontaktu. Sprowadza się on do zastąpienia cięcia dwoma połowę krótszymi cięciami prowadzonymi jednocześnie z obydwu stron kontaktu. Wówczas można sprowadzić analizę do połowy struktury z podcinanym od końca kontaktem i skorzystać ze wzoru (3.1) zastępując ξ_2" współrzędną punktu końcowego cięcia połówkowego (np. ξ" na rys.2).

3.5. Weryfikacja opisanych metod

W celu weryfikacji wyżej opisanych metod, a także potwierdzenia otrzymanych wyników zaimplementowaliśmy inną technikę numerycznego wyznaczanie rezystancji płaskich rezystorów o stałych rozłożonych opisaną w [11]. Polega ona na wyborze punktów leżących na brzegu obszaru rezystywnego i dyskretyzacji całkowego wzoru Cauche'go sprowadzonej do sum od całek pomiędzy punktami brzegowymi. Otrzymuje się w ten sposób liczbę równań liniowych dwukrotnie większą od liczby punktów na brzegu, dlatego więc użyto do rozwiązania układu równań metodę rozkładu singularnego SVD. Metoda ta jest znacznie bardziej czasochłonna (o rząd wielkości) niż opisana wyżej metoda wykorzystująca kolejne elementarne odwzorowania konforemne. Choć pozwala ona jedynie na uzyskanie dokładności około 0.2% dla ponad 100 punktów brzegowych i nie daje

żadnych możliwości konstruowania wyrażeń analitycznych aproksymujących wartość rezystancji, to jednak okazała się ona bardzo użyteczna do weryfikacji i badania dokładności opisanych wyżej metod. Uzyskano pełną zgodność wyników.

4. WYNIKI PRZEPROWADZONYCH SYMULACJI KOMPUTEROWYCH

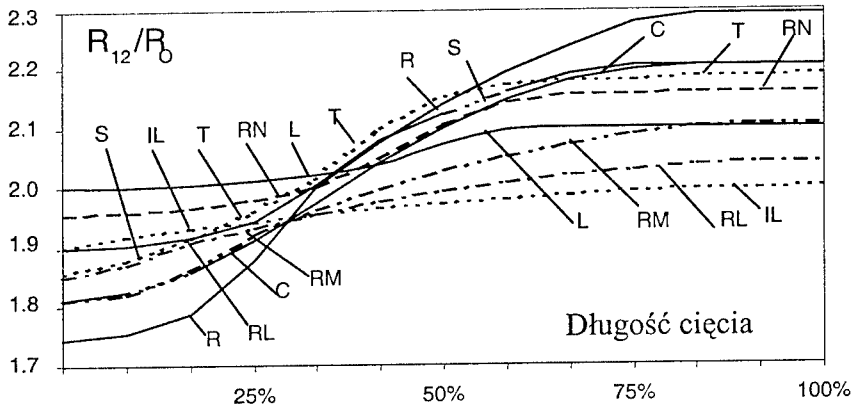
Najpierw badaliśmy zakres zmian rezystancji dla rezystora prostokątnego o stosunku boków L/D=2 dla różnych kształtów i rozmiarów rozmieszczonego symetrycznie dodatkowego kontaktu. Wyniki zestawiono w tablicy 4.1. Z tablicy tej wynika, że zarówno kształt jak i rozmiary dodatkowego kontaktu mają istotny wpływ na zakres zmian rezystancji, który przy uwzględnieniu, że początkowa wartość rezystancji bez tego kontaktu wynosiła $2R_0$ zostaje prawie podwojony.

Tablica 4.1

Zakresy korekcji dla rezystorów prostokątnych o rozmiarach L = 2D i dodatkowym kontakcie o różnym kształcie: C = półkołowym, T = trójkątnym i R = prostokątnym z B/L = 0.25 i H/D = 0.25, oraz S = R(B/2), RN = R(B/4), IN = R(B = 0), RM = R(H/2), RL = R(H/4) i IL = R(H = 0).

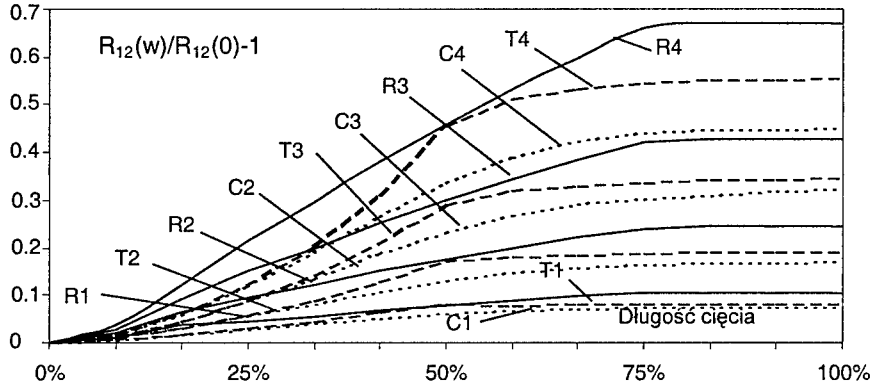
kształt kontaktu:	R	C	RM	RL	T	S	IL	RN	IN
R_{12}/R_O	1.744	1.813	1.813	1.852	1.859	1.899	1.904	1.957	2.0
R_{12C}/R_O	2.293	2.206	2.107	2.043	2.189	2.207	2.0	2.160	2.101

Następnie wyznaczyliśmy charakterystyki korekcyjne, tzn. zależność rezystancji w funkcji długości cięcia przedstawione na rys.4. Widać z nich, że choć kontakt i cięcie prostokątne zapewnia największy zakres korekcji, to jest ono niewskazane ze względu na dużą szybkość zmiany rezystancji która to wpływa na zmniejszenie dokładności korekcji. Znacznie bardziej liniową charakterystykę otrzymuje się dla kontaktów i cięć po okręgu, które jednocześnie zapewniają wystarczający zakres strojenia.



Rys.4. Zależności rezystancji (unormowanej do rezystancji powierzchniowej) analizowanych rezystorów od długości cięcia unormowanej do długości cięcia całkowicie odcinającego dodatkowy kontakt.

Był badany także wpływ szerokości cięcia na zmianę wartości rezystancji, a także wpływ odległości od kontaktu przewodzącego. Ponieważ szerokość cięcia w niewielkim stopniu wpływa na zmianę krzywizny w funkcji długości cięcia więc może być ona uwzględniona w postaci współczynnika który po dodaniu do jedyni należy przemnożyć wartość rezystancji otrzymanej dla cięcia nieskończenie cienkiego. Wartości tego współczynnika w funkcji długości cięcia przedstawiono na rys.5.



Rys.5. Współczynniki korekcyjne uwzględniające szerokość cięcia (w=i*0.0625*D) w funkcji unormowanej wartości długości cięcia dla trójkątnych (T*i), półkołowych (C*i) i prostokątnych (R*i) kształtów dodatkowego kontaktu przewodzącego.

Podobnie może być potraktowany wpływ odległości od kontaktu przewodzącego. Jeśli odległość ta jest znacznie mniejsza od rozmiarów kontaktu odpowiednie krzywe prawie się pokrywają z krzywymi przedstawionymi na rys.5 jeżeli szerokość cięcia w zamienić wielkością sumy odległości i szerokości cięcia (d+w). W odróżnieniu od zakresu korekcji współczynnik ten tylko nieznacznie zależy od rozmiarów kontaktu H i B a jedynie od jego kształtu.

6. WNIOSKI

W artykule zaprezentowano analizę nowego podejścia do korekcji laserowej rezystorów warstwowych polegającego na wprowadzeniu dodatkowego kontaktu przewodzącego który następnie jest całkowicie lub częściowo odcinany. Upraszcza to projektowanie rezystorów na wartość początkową rezystancji, rozszerza się zakres zmian rezystancji, a dzięki skróceniu długości cięcia prowadzi to do tańszego procesu produkcji układów scalonych.

Opracowano sposób wyznaczania charakterystyk korekcyjnych, a także metody uzyskiwania aproksymujących wyrażeń analitycznych koniecznych dla procesu korekcji algorytmicznej.

Na podstawie wyników przeprowadzonych symulacji doszliśmy do wniosku, że najbardziej optymalnym kształtem dodatkowego kontaktu wydaje się być półkole, zaś cięcie powinno być prowadzone w niewielkiej odległości od kontaktu do uzyskania wartości rezystancji nieco mniejszej (ok.1%) od wartości nominalnej, po czym nastąpiła by zmiana promienia i kierunku (do kontaktu) trajektorii cięcia. Zapewni to realizację korekcji przy

niewielkiej długości cięcia oraz uzyskanie dość dużej dokładności. W celu wypracowania odpowiednich zależności analitycznych i metodologii syntezy takich struktur wymagane są dalsze badania. Umożliwiłyby one na wbudowanie ich w automatyczny korektor co zaowocowałoby zwiększeniem szybkości procesu korekcji i wzrostem dokładności.

BIBLIOGRAFIA

[1] Licari J.J., Enlow L.R.: *Hybrid Microcircuit Technology Handbook. Materials, Processes, Design, Testing and Production,* Noyes Publication, New Jerssey, USA.
[2] Bube K.R., Miller A.Z., Howe A., Antoni B.: *Influence of Laser-Trim Configuration on Stability of Small Thick-Film Resistors,* W: Solid State Technology, Nov.1978, pp.55-60.
[3] Ronniger R.: *Experience with Trimming of Chip Resistors and how to Improve Trimming Results with CAD Simulation and Thick Film Process Control,* W: Hybrid Circuits, No.9, Jan.1986, pp.40-43.
[4] Chaudhry M.A.: *Design Optimisation of Integrated Circuit Thin-film Resistors with Slit and Curved Boundaries,* W: Microelectronics International, No.40, May 1996, pp.17-21.
[5] Wroński M., *Sposób korekcji wartości rezystancji rezystorów warstwowych,* Zgłoszenie patentowe P.356950 z dn. 4.11.2002, 11str.
[6] Wroński M.: *Analysis, Synthesis and Optimisation of Multicontact Distributed Resistive Networks,* W: Proc.of the 12th ECCTD, vol.2, Istanbul, Turkey, Aug.1995, pp.925-928.
[7] Filczakov P.F.: *Približennyje Metody Konformnych Otobrażenij, N.Dumka, Kijev, 1964*
[8] Wroński M.: *"Analysis and Design Optimisation of Trimmed Film Resistors"*, W: Proc.of the 21st ISHM Conf. Poland Chapter, Ustroń, Oct. 5-8, 1997, pp. 63-68.
[9] Wroński M.: *"Modelling of Trimmed Film Resistors for algorithmic laser trims"*, W: Proc.of the 22nd IMAPS Conf. Poland Chapter, Zakopane, Oct. 1-3, 1998, pp. 343-346.
[10] Wroński M.: *"Real time resistance calculation of semicircular-cut trimmed film resistors for algorithmic laser trims"*, W: Proc.of the 23rd IMAPS Conf. Poland Chapter, Kołobrzeg, Sept. 21-23, 1999, pp. 243-248.
[11] Chawala B. R., Gummel H. K.: *"A boundary technique for calculation of distributed resistance", IEEE Trans. on Electron Devices, vol. ED-17, Oct. 1970, pp. 915-925.*

NEW APPROACH TO DESIGN AND REALIZATION OF LASER-TRIM FILM RESISTORS

Summary

In this paper a new approach to the design and realisation of laser-trim film resistors is suggested and analysed. It involves creating an additional contact around which a cut is performed. This lowers the resistance value between others contacts, i.e. the initial resistance, which means simplifying the design and widening the resistance trimming range. Furthermore, improvement of sensitive characteristics takes place, and because nominal resistance value is achieved with a smaller length of the cut (or smaller resistor dimensions), it can lead to a faster and cheaper IC fabrication process. Moreover, the additional contact improves heat removal conditions, so the maximum temperature will be lower, which in turns influences the rise of stability.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Krzysztof Bikonis, Marek Moszyński, Andrzej Stepnowski

Katedra Systemów Geoinformatycznych, Politechnika Gdańska

TRÓJWYMIAROWA WIZUALIZACJA OBIEKTÓW NA DNIE MORSKIM Z ZASTOSOWANIEM TECHNIK WIRTUALNEJ RZECZYWISTOŚCI

Streszczenie

Wśród wielu nowatorskich technik stosowanych do wizualizacji danych pomiarowych coraz większą rolę odgrywają te oparte na językach opisu wirtualnej rzeczywistości. Język VRML (ang. *Virtual Reality Modeling Language*), należący do grupy języków opisu wirtualnej rzeczywistości, stosowany jest z powodzeniem w tak różnych dziedzinach jak chemia, medycyna, geografia, astronomia i nawigacja. Głównym celem artykułu jest pokazanie przydatności tego języka w tworzeniu trójwymiarowego obrazu dna morskiego i obiektów leżących na dnie.
W niniejszym artykule został zaprezentowany sposób tworzenia trójwymiarowego obrazu na podstawie danych pochodzących z echosondy wielowiązkowej. Współczesne echosondy wielowiązkowe umożliwiają lokalizację celu w trójwymiarowym układzie współrzędnych, co w połączeniu z wbudowanymi mechanizmami języka VRML pozwala na wizualizację położenia wraków, min i innych obiektów znajdujących się w toni wodnej. Dodatkowo, przeglądarka WWW wyposażona w odpowiednie rozszerzenie umożliwia dystrybucje wirtualnej sceny w sieci Internet.

1. WSTĘP

W ostatnich latach szybki rozwój technologii informacyjnych umożliwił znaczący rozwój metod prezentacji danych pomiarowych wykorzystywanych w hydroakustyce i hydrografii. Rewolucyjny postęp w komputerowych technikach mapowania oraz obrazowania znacząco wpłynął na rozwój metod badania, monitorowania oraz wizualizacji między innymi skupionych i rozproszonych obiektów podwodnych, takich jak warstwy rozproszone i ławice ryb pelagicznych oraz dno morskie. Interferometry, sonary boczne, lasery sondujące głębokość i wyspecjalizowane echosondy wielowiązkowe współpracując ze sobą w różnych systemach pozwalają tworzyć obrazy relatywnie dużych obszarów dna morskiego z niespotykaną do tej pory rozdzielczością. Ilość danych zgromadzonych przez te systemy stanowi ogromne wyzwanie dla algorytmów integrujących oraz interpretujących dane, a generalnie dla systemów prezentacji danych wyzwaniem staje się ich trójwymiarowa wizualizacja w kontekście geograficznym [1].

Współczesne oprogramowanie komputerowe pracujące na coraz nowocześniejszym sprzęcie wprowadziło nie tylko nowe, ale i bardziej wydajne metody prezentacji danych.

Rozpowszechnienie się technologii *OpenGL* oraz akceleratorów grafiki trójwymiarowej w ostatnich latach, wraz z dedykowanymi narzędziami do analizy i obróbki tej grafiki, przyczyniło się do zdynamizowania rozwoju metod prezentacji danych pomiarowych. W prezentowanym artykule przedstawiono koncepcję dynamicznego trójwymiarowego obrazowania obiektów podwodnych za pomocą języków opisu wirtualnej rzeczywistości. W artykule zaproponowano algorytm przetwarzania danych pomiarowych oraz tworzenia na tej podstawie wirtualnych scen zawierających wraki statków osadzonych na dnie. Opisane oprogramowanie do tworzenia wirtualnych scen przetwarza dane sekwencyjnie, dzięki czemu sceny mogą być tworzone na bieżąco, niemalże w czasie rzeczywistym.

2. SFORMUŁOWANIE PROBLEMU

Programy do akwizycji danych gromadzące dane z systemów wielowiązkowych wyposażone są w narzędzia do prostej dwuwymiarowej wizualizacji. W dedykowanych do tego typu zadań pakietach znajdują się narzędzia do obróbki i prezentacji danych uzyskiwanych z sonarów wielowiązkowych, interferometrów, oraz sonarów bocznych. Przykładowo należy tutaj wspomnieć o pakiecie MB-system, którego rozwój jest kontynuowany głównie przez National Science Fundation (NFS), SeaBeam Instruments i NOAA [6]. Najnowszym podejściem do prezentacji danych pomiarowych jest ich trójwymiarowa wizualizacja. Specyficzne narzędzia do trójwymiarowej wizualizacji ekosystemów morskich są przeznaczone do ułatwiania integracji i interpretacji bardzo dużych i złożonych zbiorów danych wraz z ich przestrzenną wieloskładnikową prezentacją [2]. Dane z systemów wielowiązkowych mogą być także czytane przez popularne oprogramowanie naukowe (przykładowo Matlab). Typowa reprezentacja danych dla tego typu oprogramowania to trzykolumnowe macierze w kartezjańskim układzie współrzędnych *xyz*, gdzie *xy* są współrzędnymi położenia typowymi dla sytemu UTM (ang. *Universal Transverse Mercator*), natomiast z jest współrzędną reprezentująca głębokość. Współrzędne sytemu UTM wyznaczone za pomocą systemu GPS zintegrowane razem z danymi z systemem wielowiązkowym tworzą spójną całość.

Typowe zobrazowania dna morskiego za pomocą danych z systemów wielowiązkowych są tworzone bezpośrednio w wyniku przetwarzania sygnałów echa od obiektów podwodnych, zwykle w kolejnych transmisjach sonaru lub echosondy, ze skalą kolorów reprezentującą głębokość. Macierze reprezentujące prawie regularną siatkę mogą być zorganizowane w kolorowe powierzchnie tworzące obraz dna morskiego. Powierzchnie te mogą być przetwarzane za pomocą takich technik, jak triangulacja i decymacja dla osiągnięcia kompromisu pomiędzy dokładnością (rozdzielczością), a szybkością wyświetlania otrzymywanego obrazu komputerowego. W przypadku wizualizacji obrazów z sonarów wielowiązkowych zawierających obiekty osadzone na dnie, bezpośrednia ich wizualizacja prowadzi do artefaktów związanych z łączeniem sąsiadujących punktów w powierzchnie.

3. IMPLEMENTACJA

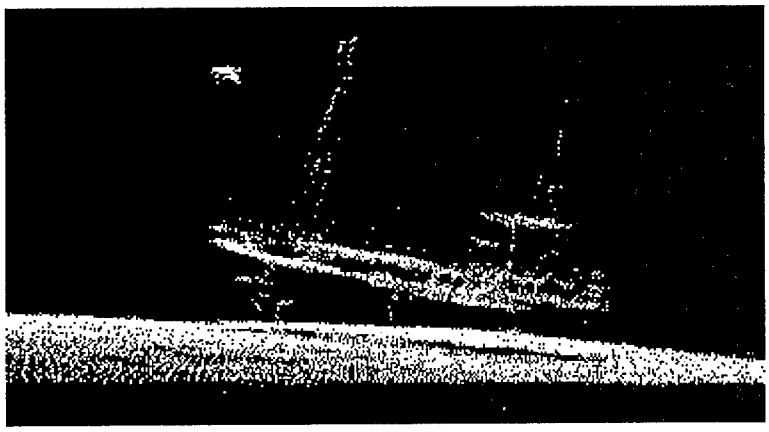
Dane pochodzące z systemów wielowiązkowych mogą być w stosunkowo łatwy sposób osadzone w prostej trójwymiarowej scenie poprzez wykorzystanie własności języka VRML. W kilku liniach kodu z użyciem węzła `PointSet` VRML pozwala na stworzenie prostej trójwymiarowej sceny przedstawionej na rysunku 1. Przedstawiony obiekt repre-

zentowany jest przez jednokolorowe punkty. Poprzez dodanie innego węzła zawierającego informacje o kolorze każdego z punktów z węzła `PointSet` uzyskuje się trójwymiarowe zobrazowanie przedstawiające kolejne echa za pomocą sekwencji punktów, natomiast głębokość za pomocą kolorów. Takie zobrazowanie może nie być satysfakcjonujące, ponieważ nie zawiera informacji, które z punktów tworzą wspólnie powierzchnie zamkniętą.

a)

```
#VRML V2.0 utf8
Shape {
        Geometry PointSet {
                Coord Coordinate {
                        Point [
                                ...
                                współrzędne xyz z danych akustycznych
                                ...
                        ]
                }
        }
}
```

b)

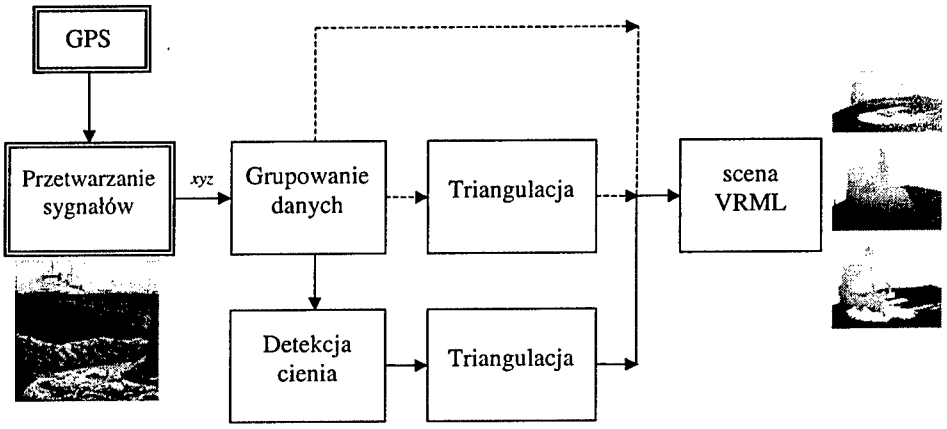


Rys. 1. Przykład trójwymiarowej sceny zawierającej wrak statku
a) fragment kodu źródłowego w języku VRML,
b) zrzut ekranu.

W przypadku obrazowania samego dna bez obiektów poprzez drobną modyfikacje kodu VRML można umieścić w scenie dodatkową informacje, o tym które z punktów tworzą powierzchnie zamkniętą. W tym celu można wykorzystać węzeł `ElevationGrid`, lub kombinację węzła `IndexedFaceSet` z węzłem `IndexedLineSet`.

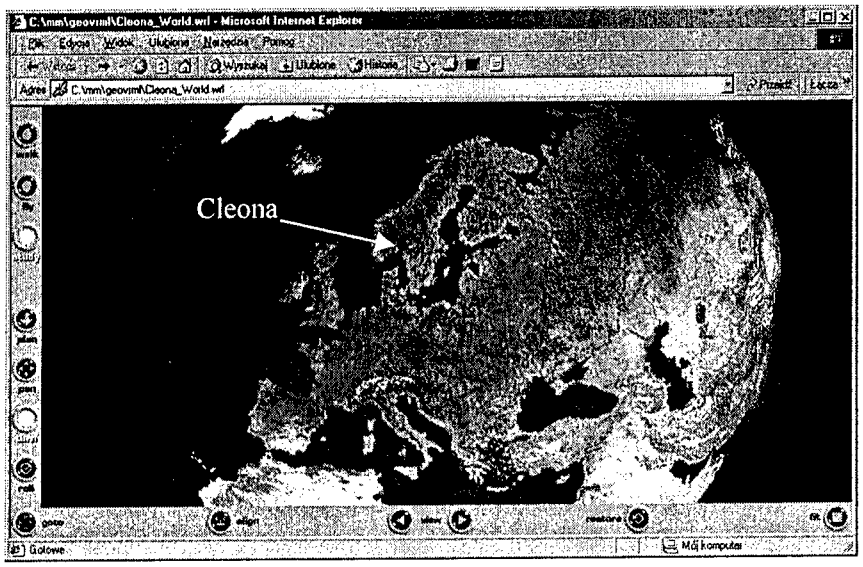
W przypadku, gdy na dnie morza znajdują się nietypowe obiekty (np.: wraki, miny) w celu ich identyfikacji wskazane jest zastosowanie specjalnego algorytmu przetwarzania danych. Schemat blokowy algorytmu przedstawiono na rysunku 2. Najprostszą wersję działania algorytmu przedstawiono linią przerywaną na schemacie blokowym, w której dane analizowane są sekwencyjnie ping po pingu. Taki algorytm zaimplementowany w systemie pomiarowym może analizować i wizualizować dane w czasie rzeczywistym. Chociaż dane potrzebne do wizualizacji otrzymywane są prosto z systemu pomiarowego,

nie zawierają informacji, które z nich tworzą powierzchnie zamkniętą. Dlatego przed triangulacją następuje proces decyzyjny, które z ech mogą utworzyć takie powierzchnie. Algorytm taki zawarty jest w bloku „detekcja cienia", w którym następuje usunięcie niektórych powierzchni zamkniętych. Czynnikiem decydującym o tym, które powierzchnie mogą zostać usunięte jest odległość pomiędzy kolejnymi punktami.



Rys. 2. Schemat blokowy tworzenia wirtualnej sceny.

Tak przetworzone dane wraków mogą teraz zostać zanurzone w scenie VRML z wykorzystaniem współrzędnych geograficznych. Do tego zadania została wykorzystana technologia GeoVRML, która pozwala prezentować w sposób trójwymiarowy dane geograficzne [3].



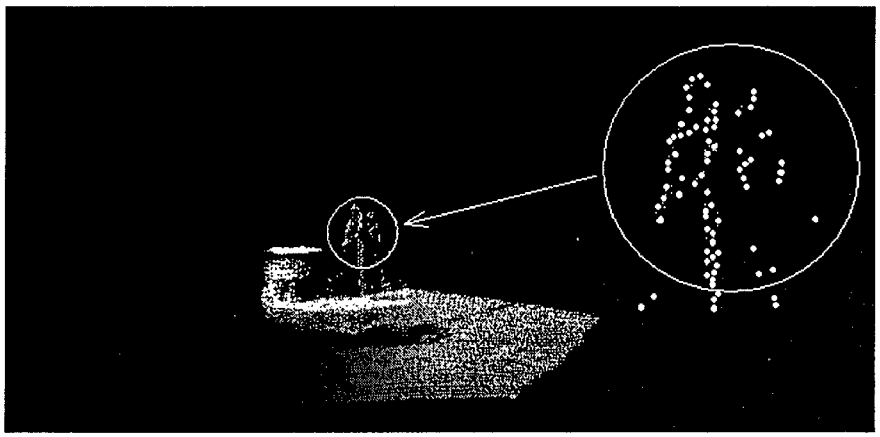
Rys. 3. Lokalizacja wraku Cleony w Zatoce Oslo za pomocą techniki GeoVRML.

GeoVRML jest techniką dostarczającą rozwiązania, które pozwalają na reprezentację oraz wizualizację danych z wykorzystaniem standardu VRML97 (standard trójwymiarowej prezentacji danych w sieci Internet) poprzez zdefiniowanie węzłów dedykowanych danym geograficznym. Rezultat uzyskany dzięki tej technice prezentuje rysunek 3. Wybór punktu symbolizującego wrak pozwala na szczegółowe wyświetlenie obrazu wraku. Innym sposobem na szczegółowe wyświetlenie obrazu wraku jest zastosowanie węzła LOD (ang. *Level of Detail*). Pozwala on na zdefiniowanie skali szczegółowości wyświetlanych elementów w scenie w funkcji ich odległości od obserwatora.

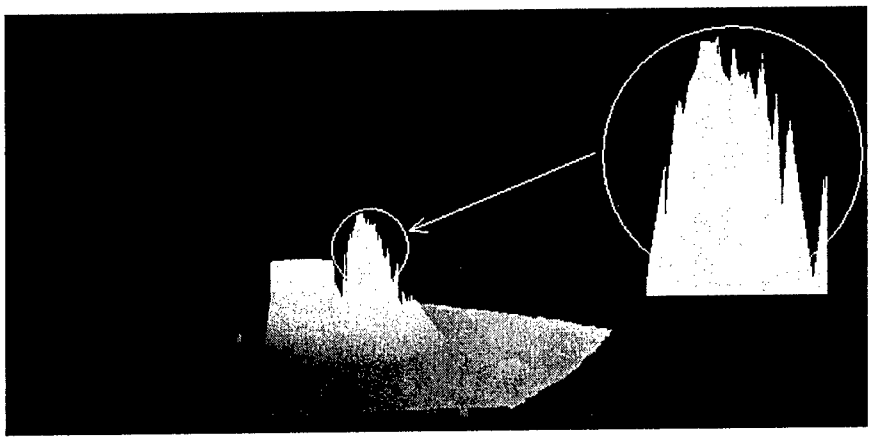
4. WYNIKI

W celu weryfikacji zaproponowanego algorytmu wykorzystano przykładowe dane o wraku pochodzące z przeszukiwań akustycznych wykonanych w Zatoce Oslo przy pomocy sonaru wielowiązkowego SEABAT. Dane te otrzymano w ramach współpracy z duńską firmą Reson, która jest producentem tego typu sonarów. Przedstawiają one wrak statku Cleona, który został zwodowany w Borhaug w 1918 roku i służył do transportu drewna. Po wojnie funkcjonował jako jednostka transportowa w Zatoce Oslo. Zatonął w okolicach Leangbukta i spoczywa na dnie na głębokości piętnastu metrów w relatywnie dobrym stanie.

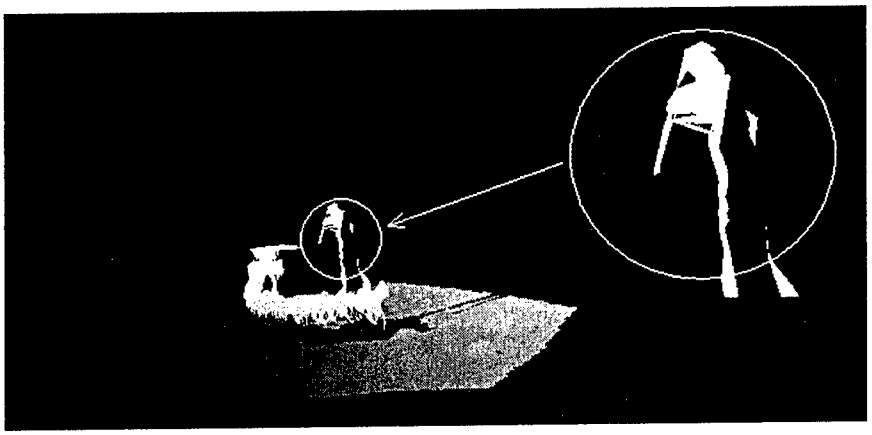
Na rysunkach od 4 do 6 przedstawiono zrzuty ekranu trójwymiarowych scen prezentujących wyniki uzyskane za pomocą proponowanych algorytmów. Rysunek 4 prezentuje trójwymiarową scenę zawierającą wrak zobrazowany za pomocą punktów reprezentujących kolejne echa. Taka forma prezentacji może być wykorzystana, gdy obserwator znajduje się niezbyt blisko obiektu. Rysunek 5 pokazuje typowy obraz wraku po automatycznym połączeniu punktów w powierzchnie zamknięte za pomocą triangulacji. Rysunek 6 przedstawia rezultat uzyskany z zastosowaniem algorytmu detekcji cienia. Przedstawia on tylko te powierzchnie, które po detekcji cienia zostały zakwalifikowane, jako te reprezentujące rzeczywiste powierzchnie.



Rys. 4. Kolorowa prezentacja Cleony za pomocą węzła `PointSet`.



Rys. 5. Kolorowa prezentacja Cleony za pomocą węzła `IndexedFaceSet` po triangulacji.



Rys 6. Kolorowa prezentacja Cleony za pomocą węzła `IndexedFaceSet` po triangulacji oraz algorytmie detekcji cienia.

5. PODSUMOWANIE

Dotychczas stosowane metody akwizycji i prezentacji danych pomiarowych pochodzących z przeszukiwań hydroakustycznych pozwalały na różnorakie formy dwuwymiarowej wizualizacji różnych obiektów podwodnych i całych obszarów hydrosfery (włączając w to również dno morskie). Dwuwymiarowe obrazowania typu A, B, M i P [5] uzyskiwano w wyniku bezpośredniego przetwarzania sygnałów echa od obiektów podwodnych pochodzących z kolejnych transmisji sonaru lub echosondy. Przełomem w wizualizacji obiektów podwodnych może stać się wprowadzenie języków opisu wirtualnej rzeczywistości. Jak pokazano szczególnie przydatny wydaje się tu być język VRML, który wraz z szybkim rozwojem technik informacyjnych obok typowych aplikacji w grafice komputerowej coraz chętniej wykorzystywany jest w wielu różnych dziedzinach.

BIBLIOGRAFIA

[1] Hansen R. K.: *Applications of 3D real time acoustical imaging*, Underwater Intervention, New Orlans, 1999.

[2] Gee L., Paton M., Mayer L. A., Ware C., Reed B., Depner J.: *Interactive 3D visualization and analisys of multi-sensor, multi-resolution data sets*, Proceedings of The 3rd International Conference on High Resolution Surveys in Shallow Water, Sydney, Australia, November 17-20, 2003.

[3] Reddy M., Iverson L., Leclerc Y. G.: *Under the Hood of GeoVRML 1.0*, Proceedings of The 5th Web3D/VRML Symposium, Monterey, California, February 21-24, 2000.

[4] Moszyński M., Bikonis K., Stepnowski A.: *Dynamic 3D visualization of fish movement from split-beam data*, Proceedings of The Congres on Ultrasonic 2003, Paris, France, September 7-10, 2003.

[5] Bikonis K., Moszyński M.: *Wizualizacja w języku VRML przemieszczania się ryb pelagicznych w wiązce echosondy*, Zeszyty Naukowe ETI Politechniki Gdańskiej, seria: Technologie Informacyjne, 2003, I Krajowa Konferencja Technologie Informacyjne, Gdańsk, 18-21 Maj, 2003.

[6] Caress D.W., Chayes D.N., materiały dostępne w wersji Internetowej pod adresem http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbsystem_home.html

3D VISUALIZATION OF OBJECTS ON SEA FLOOR USING VIRTUAL REALITY TECHNIQUES

Summary

Virtual Reality Modeling Language VRML despite the typical application in computer graphics is also used in various fields. For instance in chemistry it is used for visualization of atom particles, in medicine for modeling of human organs, in astronomy for visualization of trajectories of the stars and in geography and navigation for creation of 3D maps.

In the paper VRML is used for visualization of ship wrecks located on the sea floor. Developed algorithm creates 3D active images from acoustic data acquired by a multibeam sonar. As the modern multibeam echosounders and sonars allows for unambiguous localization of seafloor targets, it is

possible to visualize wracks, mines and other submerged objects in 3D coordinate system using built-in properties of VRML. Additionally, as the WWW browser can be equipped with VRML plugin 3D visualization of objects can be redistributed in the Internet.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Aldona Chlewicka

Katedra Bibliotekoznawstwa i Informacji Naukowej, Akademia Bydgoska

ŚREDNIOWIECZNE RĘKOPISY W ZASOBACH ŚWIATOWEGO INTERNETU

Streszczenie

Zasoby Internetu dotyczące średniowiecznych rękopisów są bardzo obszerne. Oprócz pojedynczych ilustracji pochodzących ze znanych rękopisów, znaleźć można całe biblioteki zdigitalizowanych manuskryptów, ich krytyczne edycje wraz z aparatem naukowym, bibliograficzne bazy danych. W tworzeniu baz największą rolę odgrywają biblioteki, głównie narodowe i placówki naukowe, choć nie brak również stron projektowanych przez zapalonych amatorów. Adresatami stron są głównie osoby profesjonalnie zajmujące się dawną książką: bibliolodzy, historycy sztuki, mediewiści, jak również studenci i miłośnicy dawnej książki.

1. WSTĘP

Współczesny badacz historii książki i bibliotek ma dziś do dyspozycji prawie nieograniczone możliwości w docieraniu do źródeł informacji. Obok tradycyjnych tekstów drukowanych ma do dyspozycji wspaniale wydane z dbałością o najwyższą jakość kolorowe albumy, filmy na kasetach VHS i płytach DVD oraz bogate źródła elektroniczne i multimedialne[1]. Jednak największe możliwości daje dziś z pewnością Internet, choć jak zwykle jednym z większych problemów staje się wyszukanie i dotarcie do odpowiadającej mu informacji. Z pewnością najpełniejsze i najbardziej wiarygodne dane pochodzą od instytucji profesjonalnie zajmujących się problematyką książki i bibliotek. Do nich należą głównie duże, naukowe biblioteki oraz instytuty związane z propagowaniem wiedzy z zakresu bibliologii i informacji naukowej. Po rozwiązaniu najbardziej palących problemów związanych z automatyzacją pracy w bibliotekach, wdrożeniem programów do obsługi procesów bibliotecznych, konwersją i retrokonwersją katalogów i ich dostępem *online*, wdrożeniem modułów umożliwiających zamawianie i dostęp do publikacji poprzez Internet, zainteresowano się w bibliotekach innymi przedsięwzięciami.

[1] O internetowych źródłach w badaniach nad bibliotekami i ich zbiorami pisała m.in. Maria Kocójowa, Zob. Tejże: *Internet jako źródło do badań bibliotekoznawczych,* W: *Elektroniczne publikacje w bibliotekach*, red. nauk. Maria Kocójowa, Kraków: Wydaw. UJ, 2002, s. 96-107.

Duża część bibliotek skupia obecnie swoje wysiłki na prezentacji swoich najcenniejszych zbiorów[2]. Niektóre z dokumentów zawartych w kolekcjach zaliczane są do światowego dziedzictwa kulturowego. Jedną z najciekawszych, a zarazem najbezpieczniejszych dla cennych obiektów metod jest ich udostępnianie w formie zdigitalizowanej. Biblioteki na szeroką skalę stosują obecnie programy zmierzające do przeniesienia swoich cimeliów na nośniki elektroniczne. Najwartościowszą grupę wśród zbiorów bibliotecznych stanowią z pewnością średniowieczne manuskrypty, których unikatowość, artyzm przesądza na ogół o bardzo ograniczonych możliwościach ich tradycyjnego udostępniania. Większość dużych bibliotek europejskich mogących się poszczycić bogatymi kolekcjami manuskryptów z początkiem lat 90. rozpoczęła długofalowe programy zmierzające do ich digitalizacji[3]. Efektem tej działalności są obiekty elektroniczne dostępne głównie w Internecie i na CD-ROM-ach, choć możliwości prezentacji są właściwie nieograniczone. Biblioteka Brytyjska udostępnia swoim zwiedzającym książki wirtualne, które za pośrednictwem aktywnego monitora można dowolnie otwierać i wirtualnie przeglądać. Nad ekranem monitora przesuwa się rękę imitując przewracanie kartek, stronice rozchylają się i miękko spływają na bok. W tym miejscu stwierdzenie Umberto Eco, który w jednym ze swoich esejów nt. przyszłości książki napisał: ... *Nie możemy natomiast pozbyć się książek do czytania – od „Iliady" po najnowszą powieść. Nie tylko dlatego, że czytanie od „deski do deski" na ekranie jest męczące, ale i dlatego, że forma-książka to jeden z tych wynalazków* [...], *dla których nie znaleziono w ciągu stuleci lepszych pod względem ergonomicznym odpowiedników. Książka dobrze pasuje do ręki...* wydaje się być wciąż aktualne[4]. Nasze przywiązanie do tradycyjnej formy książki trudne jest do przezwyciężenia, choć już dziś w głównie w naukach ścisłych utrwala się pogląd, *że jeśli coś nie istnieje w Internecie nie istnieje w ogóle*[5].

2. INTERNETOWE ZASOBY INFORMACJI O ŚREDNIOWIECZNYCH RĘKOPISACH

Na czele instytucji, które skupiły swoje wysiłki na realizacji projektów dążących do przekształcenia swoich tradycyjnych zbiorów w cyfrowe zasoby stoją niewątpliwie biblioteki narodowe. Część z nich realizuje swoje prace, w sposób bardziej zaawansowany, od początku lat 90. O ile w pierwszym etapie programy te nie miały wypracowanej dalekosiężnej strategii uwzględniającej zasięg, przyszłość i zmianę nośników itp. i były przedsięwzięciami, których celem było przede wszystkim zaprezentowanie swoich najcenniejszych zasobów w Internecie, o tyle w obecnej fazie projekty te są już bardziej zaawansowane i obejmują kompleksowe zagadnienia polegające na prezentacji cyfrowych obrazów średniowiecznych rękopisów przechowywanych w całej sieci bibliotek. W tym celu zawiązuje się współpraca od kilku do kilkudziesięciu bibliotek przechowujących manuskrypty i instytucji patronujących i finansujących w dużej mierze te przedsięwzięcia.

---

[2] Obok zbiorów rękopiśmiennych, bardzo dobrze reprezentowane są w Internecie stare druki. Zob. Katarzyna Ślaska, *Stare druki w Internecie. Najciekawsze strony www związane z dawną książką,* Biuletyn Informacyjny Biblioteki Narodowej, 1993, nr 3, s. 31-35.

[3] Grupie tej przewodzą biblioteki narodowe. Zob. np. Anne Amse, *Zabezpieczanie historycznych zasobów dla przyszłości - archiwum cyfrowe holenderskiej biblioteki narodowej,* Biuletyn EBIB [Czasopismo elektroniczne] 2003, nr 2. Tryb dostępu: http://ebib.oss.wroc.pl/2003/42/amse.php.

[4] Umberto Eco, *Czy komputer pożre książkę?,* przeł. Adam Szymanowski, Gazeta Wyborcza, 1996, nr ,s. 8-9.

[5] David Pearson, *Digitalizacja - czy istnieje jakaś strategia?* Biuletyn EBIB [Czasopismo elektroniczne] 2003, nr 2. Tryb dostępu: http://ebib.oss.wroc.pl/2003/42/pearson.php

Z pewnością do najbardziej zaawansowanych należy francuski blok projektów powstających w ramach współpracy z IRHT (Institut de Recherche et d'Histoire des Textes). Jednym z głównych przedsięwzięć jest baza: **Enluminures**[6] zawierająca 14000 cyfrowych obrazów pochodzących z 1200 rękopisów francuskich bibliotek miejskich. Stworzenie bazy było możliwe dzięki patronatowi ministerstwa kultury, które jest odpowiedzialne za digitalizację francuskich zbiorów muzealnych, bibliotecznych, archiwalnych, fotograficznych, rękopiśmiennych i in. oraz Direction du Livre et de la Lecture. Baza jest wciąż rozwijana. Obecnie należy do niej ponad 60 bibliotek, m. in. w Lyonie, Poitiers, Grenoble, Lille, Mans, Metz, Soissons. Ostatnio wzbogaciła się o wirtualna ekspozycję *Le moyen age en lumiere*, która umożliwia oglądanie wybranych stron z bogato zdobionych rękopisów w dużym powiększeniu. Możliwe jest wyświetlenie tekstów i informacji również w języku niemieckim i angielskim. W bazie tej tematy ikonograficzne są pogrupowane. Obecnie dostępne są m. in. *czas i kosmos, dzień i praca, człowiek i rodzina, wyobrażenia Boga.* Wybrane obrazy, wysokiej rozdzielczości, otwierają się podczas prezentacji w nowym oknie.

Następny z projektów francuskich to **Liber Floridus**[7]. Jest to baza danych o rękopisach przechowywanych w dwóch bibliotekach: Biblibilotheque Sainte-Genevieve i Bibliotheque Mazarine. Baza opracowana została przez wymienione biblioteki jak również CINES (Centre Informatique National de l'Enseignement Superieur) oraz IRHT[8], we współpracy z Mnistre de la Jeunesse, de l'Education nationale et de la Recherche. Baza, umieszczona na serwerze IRHT, zawiera ponad 15 tys. obrazów z 2383 rękopisów z biblioteki Sainte-Genevieve i ponad 17 tys. z 1280 rękopisów z biblioteki Mazarina. Wyszukiwanie manuskryptów możliwe jest poprzez dwie opcje: miejsce przechowywania i ilustracje. W pierwszym przypadku wyszukuje się rękopis w bazie wypełniając wybrane pola opisu: nazwę biblioteki, autora, tytuł, datę powstania, rodzaj piśmiennictwa (np. Biblia, liturgika, teologia moralna, duchowa), miejsce powstania, pierwszego właściciela, artystę bądź też szkołę. W tej bazie możliwe jest też odnalezienie potrzebnej ilustracji poprzez podanie jej tytułu lub słowa kluczowego np. „vie militaire" lub „soldat". Znakomitym ułatwieniem są umieszczone przy każdym polu indeksy, które tak jak i opisy i ilustracje otwierają się w osobnym oknie. Gdy wybrana zostanie więcej niż jedna ilustracja np. o określonej tematyce, wszystkie widoczne są w nowym oknie, przy czym pierwsza umieszczona jest w powiększeniu wyżej. Nie trzeba ponownie wpisywać słowa kluczowego, aby przejść do następnej ilustracji.

Do większych projektów o zasięgu ponad narodowym należy niemiecka baza **Manuscripta Mediaevalia**. Projekt ten ma umożliwiać dotarcie do informacji o rękopisach średniowiecznych przechowywanych głównie w Niemczech i w Austrii, a także w innych krajach. Jest wynikiem działań podjętych w 1996 r. w ramach projektów: „Entwicklung und Bereitstellung des Prototypes für eie Datenbank mittelalterlicher Handscxhriften" i „Starhilfe für die Zentralredaktion mittelalterlicher Handschriftenkataloge". Są to właściwie trzy powiązane ze sobą za pomocą hiperłączy bazy aktualizowane na bieżąco: Handschriften, Katalog i Datenbank. Bazę pierwszą tworzy 89 zdigitalizowanych rękopisów pełnych i fragmentów pochodzących m.in. z bibliotek w Berlinie, Darmstadt, Heidelbergu, Marburgu, Fuldy. Obrazy można oglądać powiększone 2 i 3-krotnie. Najciekawszym jest tu z pewnością Gero-Codex, proponowany na listę „Pamięć Świata" (Memory of the World"). Najważniejsza jednak jest druga baza, która zawiera blisko 60

[6] Enluminures http://www.enluminures.culture.fr/documentation/enlumine/fr/
[7] Liber Floridus http://liberfloridus.cines.fr
[8] Institut de Recherche et d'Histoire des Textes http://www.irht.cnrs.fr

tys. opisów manuskryptów z 256 katalogów tradycyjnych: drukowanych i rękopiśmiennych, które zostały zdigitalizowane i dostępne są w Internecie. Przeszukiwanie możliwe jest poprzez indeks topograficzny (wg miejsca przechowywania rękopisów) lub też chronologicznie (najstarszy katalog pochodzi z 1887 r.). Oprócz tego w bazie dostępne są także informacje o katalogach z 20 ośrodków zagranicznych m.in. Bratysławy, Krakowa, Londynu. Tylko część z tych katalogów dostępna jest w wersji cyfrowej. Ostatnia baza jest najbardziej rozbudowana. Oferuje możliwość wyszukiwania 65 tys. rękopisów w bazie wg następujących kryteriów: miejsce przechowywania, data, sygnatura, autor i inni współtwórcy, słowo w tytule, tytuł lub incipit, słowo kluczowe, proweniencje, oraz temat ilustracyjny i in. Wyniki wyszukiwań, także z zastosowaniem operatorów Boole'a można następnie wyświetlić w postaci skróconej listy (Ergebnisliste) bądź rozbudowanej, uwzględniającej pełniejszy opis dokumentu (Vollansicht) lub też przedstawiającej strukturę dokumentu tj. układ tekstu, miniatur itp. (Strukturansicht). W przyszłości ma również zostać uruchomiony tezaurus oferujący możliwość wydajniejszego przeszukiwania baz. Ostatnia z nich umożliwia wyszukanie rękopisów poprzez tematykę ich ilustracji według klasyfikacji zaprezentowanej przez system ICONOCLASS.

Spośród przedsięwzięć podjętych przez biblioteki naukowe z pewnością na uwagę zasługuje **Celtic and Medieval Manuscripts (or Early Manuscripts Project) at Oxford** stanowiący część projektu poświęconego wyspecjalizowanym badaniom zbiorów w humanistyce Specialised Research Collections in Humanistic[9] sponsorowanego przez Fundację Wyższej Edukacji Anglii, Szkocji i Płn. Irlandii. Program ten realizowany był w latach 1995-2000. Polegał na digitalizacji zasobów rękopiśmiennych bibliotek wchodzących w skład Oxford University. Do projektu wybrano 80 rękopisów z następujących kolekcji: Balliol College, Bodleian Library, Corpus Christi College, Jesus College, Magdalen College, St. John's College. Rozpoczęto od digitalizacji rękopisów celtyckich z okresu 9-19 w. Wybrano te, które ze względu na stan zachowania i swoją wartość nie mogły być prezentowane w tradycyjny sposób. Nie wszystkie manuskrypty były kompletne, uznano jednak słusznie, że lepiej ukazać dzieło we fragmencie niż wcale.

Większość obrazów powstała z zeskanowania już istniejących mikrofilmów 35 mm. Do skanowania w rozdzielczości 2700 dpi użyto aparatu Nikon LS-2000. Zdigitaliowane obrazy w formacie TIFF zajmują od 15 do 30 Mb (niekiedy więcej). Po skompresowaniu, obrazy w formacie JPEG dostępne są w następujących wielkościach:

small	500 pixeli	50 – 100 Kb
medium	1000 pixeli	200 – 350 Kb
large	1500 pixeli	500 – 700 Kb[10]

Po wczytaniu strony głównej projektu można wybrać jedną z sześciu kolekcji a następnie konkretny rękopis. Posługując się miniaturą w lewym rogu ekranu można przewracać karty rękopisu. Zatwierdzona strona otwiera się na ogół wolno ze względu bardzo dużą pojemność obrazu wyjściowego. Z lewej strony ekranu widoczna jest skala w centymetrach, która ułatwia umiejscowienie fragmentu strony. Tak duże powiększenie umożliwia bardzo dokładne zapoznanie się z tekstem, zdobieniami, stanem zachowania rękopisu. Obrazy uzupełnione są informacjami o rękopisie: sygnaturach, nr kart. Bardzo szczegółowo omówione są zagadnienia ochrony praw autorskich.

[9] Celtic and Medieval Manuscripts or (Early Manuscripts Project) at Oxford Zob. http://image.ox.ac.uk/
[10] Ibidem.

Na nieco mniejszą skalę podjęte zostały w 1994 r. działania mające na celu digitalizację a następnie opublikowanie dwóch najcenniejszych rękopisów będących w posiadaniu Biblioteki Uniwersytetu w Aberdeen. Dzięki dofinansowaniu tej samej instytucji, która wspomagała projekt oksfordzki, przy współpracy z Departamentem Historii Sztuki i Centre for Computer Based Learning in Land Use and Environmental Sciences (CLUES) Uniwersytet opracował specjalny program którego efektem jest dostęp *online* do dwóch manuskryptów: XV-wiecznego – The Burnet Psalter i XIII-wiecznego – **Aberdeen Bestiary**. W 1996 r. ukazała się pierwsza edycja pilotażowa Bestiariusza[11], a w 2002 r. następna[12]. Przedsięwzięcie obejmowało kompleksową usługę, polegającą nie tylko na digitalizacji i umieszczeniu stron w Internecie, ale także na zaopatrzeniu obrazów w niezbędne komentarze, transkrypcje i transliteracje. Projekt nie pretendował do krytycznego wydania tekstu Bestiariusza. Nowsza baza[13], zawiera kilka odsyłaczy, m.in. wprowadzenie, informacje o rękopisie, jego historię, komentarze dotyczące tekstu i zdobnictwa, indeks, informacje dotyczące transkrypcji i transliteracji, bibliografię oraz informacje dotyczące kompozycji strony. Na uwagę zasługują z pewnością komentarze kodykologiczne, najbardziej rozbudowane. Zawierają one bardzo szczegółowe informacje dotyczące budowy kodeksu np. materiału z jakiego rękopis sporządzono, wymiary, ilość składek i sposób ich oznaczania. Omówiono równie drobiazgowo liniowanie i sposób zaznaczania marginesów (za pomocą nakłuć), jak również rodzaje zdobień: rubrykowanie[14], inicjały oraz następstwo czynności przy pracy nad rękopisem. Obszerniejszą analizę przeprowadzono także nad sposobem kopiowania ilustracji tj. poprzez nakłuwanie linii gotowego rysunku, a następnie posypywanie rozdrobnionym proszkiem, który osadzał się w miejscach nakłuć[15]. Każde omówienie zawarte w tej bazie zilustrowane jest przykładem z rękopisu. Prowadzą do tego bardzo wygodne odsyłacze. Większość stron rękopisu można wyświetlić w wygodnym powiększeniu.. Komentarze w jakie zaopatrzone są obrazy stron manuskryptu przynoszą analizę znaczenia tekstu i zdobień. Bardzo łatwo od komentarza można przejść do transliteracji tekstu na język angielski. Powiązania pomiędzy stronami w bazie i przejścia pomiędzy nimi objaśnia specjalna tabela. Dzięki tym hiperlinkom można wertować rękopis przeskakując od komentarza do tłumaczenia.

Odmienne od wyżej wymienionych cele i zadania ma projekt **Pierce Plowman Electronic Archive**[16]. Jest to elektroniczna edycja 14 w. alegoryczno-satyrycznego dzieła Wlliama Laylanda. Projekt, którego początki sięgają jeszcze połowy lat 80. skierowany jest do studentów, amatorów, badaczy i krytyków, historyków literatury, mediewistów, językoznawców i osób zainteresowanych produkcją książki. Bazą źródłową programu digitalizacyjnego, szczegółowo opisanego na stronach internetowych[17] były zachowane rękopisy i ich późniejsze wydania krytyczne. Nad zapisem tekstów do języka SGML w kodzie ASCII pracował zespół. Stopniowo dodano do cyfrowego tekstu wszelkie informacje niezbędne dla jego prawidłowego odczytywania. tj. o cechach fizycznych,

[11] Aberdeen Bestiary Project - University of Aberdeen http://www.clues.abdn.ac.uk:8080/bestiary-old/alt/

[12] http://www.clues.abdn.ac.uk:8080/besttest/firstpag.html

[13] Aberdeen Bestiary Project - University of Aberdeen http://www.abdn.ac.uk/bestiary/

[14] Zaznaczanie pewnych fragmentów w tekście poprzez podkreślenie lub zapisanie odmiennym kolorem, najczęściej czerwonym.

[15] http://www.abdn.ac.uk/bestiary/codicology.hti

[16] Pierce Plowman Electronic Archive http://jefferson.village.virginia.edu/piers/tcontents.html. Projekt ten jest jednym z wielu przygotowanych przez Institute for Advanced Technology in Humanitides na Uniwersytecie w Wirginii.

[17] Zob. Hoyt N. Duggan, *Creating an Electronic Archive of Piers Plowman*, Wirginia: IATH, 2001. Tryb dostępu: http://jefferson.village.virginia.edu/piers/tcontents.html.

glosach interlinearnych i marginalnych, zmianach ręki piszącego, naniesionych korektach i skreśleniach, suspencjach i kontrakcjach, itp. Do kolacjonowania tekstów, tj. zestawienia słowo po słowie i wiersz po wierszu posłużył program COLLATE. W efekcie wyłoniły się trzy wersje tekstu: A,B,C. Projekt właściwie do dziś nie jest w pełni ukończony (ostatnia modyfikacja w sierpniu 2001 r.). Rękopisy, które składają się na tekst wyjściowy docelowo mają być dostępne w następujących opcjach: zawartość, język, opis, faksymile. Większość do tej pory ma opracowany jedynie tekst i pojedyncze obrazy[18].

Bogate zasoby wiadomości o dawnych rękopisach dostępne są także na stronach poświęconych wyłącznie zagadnieniom mediewistycznym. Są one przygotowywane przez osoby i instytucje profesjonalnie zajmujące się historią średniowiecza, czasem przez amatorów. Jednym z takich serwisów jest **Ménstrel** (Médievistes sur l'internet)[19], założony w 1997 r. z inicjatywy Urfis (Unité Régionale de Formation à l'Information Scientifique et Technique Paris - Ecole Nationale des Chartes) dla dokumentowania studiów nad średniowieczem. Serwis prezentuje informacje o bieżących wydarzeniach: konferencjach, wystawach, szkoleniach. Główne linki prowadzą do francuskich i zagranicznych bibliotek państwowych i kolekcji prywatnych, archiwów, organizacji takich jak np. IRHT, czy ARTEM (Atelier de Recherche sur les Textes Médiévaux), serwisów bibliograficznych, projektów udostępniania zbiorów średniowiecznych *online*, stron zawierających faksymile, tekstów poświeconych zagadnieniom metodycznym np. formie prezentacji rękopisów w Internecie. Szczegółowe wyniki badań prezentowane są w licznych blokach problemowych np. dyplomatyka, heraldyka, słowniki i encyklopedie (dostępne *online*), teatr średniowieczny, paleografia i manuskrypty. Ten ostatni zawiera odnośniki do ok. 200 stron, m.in. towarzystw i instytucji skupiających swoją problematykę na badaniach dawnego pisma i iluminatorstwie, ośrodków kształcących w tym zakresie, seminariów, wykładów, warsztatów, studiów doktoranckich, serwisów i projektów obejmujących np. digitalizację katalogów rękopisów. Z pewnością najważniejszą rolę odgrywają dostępne *online* teksty i artykuły, pisane głównie przez osoby profesjonalnie zajmujące się paleografią.

Na koniec chciałabym przedstawić z kolei mniej profesjonalną, a bardziej spontaniczną inicjatywę campusu KFKI (Központi Fizikai Kutató Intézet – Central Research Institute for Physics) Węgierskiej Akademii Nauk[20]. Wśród kilku znajdujących się tam baz jest też **Web Gallery (Fine Arts in Hungary)** zawierająca cyfrowe obrazy dzieł sztuki, w tym także rękopisów średniowiecznych i renesansowych umiejscowionych w bazie Illuminated manuscrits[21]. Można je przeglądać poprzez indeks artystów lub wybierając ramy czasowe. Obrazy (jeden lub kilka stron z danego rękopisu) otwierają się w nowym oknie tak jaki i podstrony zawierające mniej lub bardziej dokładne opisy. Miniatury dostępne są w różnych rozmiarach powiększeń (od 25% do 200%). Witryna campusu dostosowująca się do upodobań młodych ludzi zawiera także licznik odwiedzin (od 1996 r. ponad 2 mln odwiedzających) i opcję pozwalającą na wysyłanie kartek z życzeniami. Strona niestety dość dawno nie była aktualizowana.

Jeśli nie wystarcza nam oglądanie dawnych rękopisów na ekranie można skorzystać z usług instytucji przygotowujących ww. projekty. Większość z nich oprócz darmowych usług oferuje także płatne, polegające na sporządzeniu faksymiliów, powiększeń, foto-

[18] Zob. Hoyt N. Duggan, *Creating an Electronic Archive of Piers Plowman*, Wirginia: IATH, 2001. Tryb dostępu: http://jefferson.village.virginia.edu/piers/tcontents.html.
[19] Ménstrel. Médievistes sur l'internet : http://www.ccr.jussieu.fr/urfist/mediev.htm.
[20] KFKI (Central Research Institute for Physics) http://www.kfki.hu/campus.
[21] Illuminated manuscripts (miniatures) from the 12th to 16th centuries http://www.kfki.hu/arthp/index1.html

grafii, mikrofilmów, lub też CD-ROM-ów. Jeden z najpiękniejszych wczesnośredniowiecznych rękopisów – iryjska „Księga z Kells" (The Book of Kells) oferowana jest przez Trinity College w Dublinie w wersji elektronicznej na CD-ROM-ie[22].

Oprócz tego w Internecie znaleźć można adresy wielu firm specjalizujących się w tworzeniu faksymiliów dawnych rękopisów, które w tym celu wykorzystują dawne technologie. Jedną z takich firm jest australijska **Finn's Fine Books** z siedzibą w Nowej Południowej Walii. Specjalizuje się w produkcji wiernych kopii dawnych rękopisów i dzieł sztuki[23]. Pod względem edytorskim ich produkty niewielu różnią się od wytworów oryginalnych naśladując ich materiały i fakturę, kolor, wielkość, zdobienia itp. Firma na swojej stronie zamieszcza oprócz katalogu dokładne informacje dotyczące stosowanych technologii. Wśród ofert tej firmy znajdują się kopie najpiękniejszych rękopisów, m.in. Bardzo Bogatych Godzinek Księcia de Berry (Tres Riches Heures du Duc du Berry) oraz wspomnianej „Księgi z Kells".

3. ZAKOŃCZENIE

Są to w zdecydowanej większości przedsięwzięcia niekomercyjne. Każdy może na ekranie monitora obejrzeć cyfrowe obrazy. Prawo autorskie na ogół zezwala także na sporządzenie jednej kopii zapisanej na twardym dysku i ewentualnego wydruku, choć niektóre projekty przewidują zabezpieczanie swoich obrazów znakami wodnymi (watermarks)[24]. Publiczne udostępnianie jest zabronione z wyjątkiem ograniczonego wykorzystania materiałów w dydaktyce i nauce. Bazy zawierające cyfrowe obrazy manuskryptów pełnią podobną funkcję w nauczaniu jak zdigitalizowane obiekty muzealne tj. służą kształceniu przyszłych bibliotekarz, mediewistów, artystów, historyków sztuki czy filologów[25]. Większość projektów wśród swoich celów wymienia właśnie służenie nauce i dydaktyce. Należy pamiętać, że ok. 60% użytkowników Internetu to osoby szukające materiałów edukacyjnych[26].

Obecne programy strategiczne dotyczące digitalizacji iluminowanych manuskryptów opracowywane są na ogół na poziomie poszczególnych krajów, przy czym wiodącą rolę odgrywają tu biblioteki narodowe. Z pewnością kolejne lata przyniosą rozwój strategii, w których planowanie i realizacja przedsięwzięć odbywać się będzie na poziomie międzynarodowym. Jest nadzieja, że programy te będą w dalszej części realizowane na zasadzie non-profit i nie zdominują je przedsięwzięcia komercyjne jak to stało się z elektronicznymi czasopismami.

---

[22] Trinity College Dublin http://www.tcd.ie/Library/Shop/catalog.php

[23] Finn's Fine Books http://finns-book.com

[24] Zob. Irish Script on Screen : http://www.isos.dcu.ie

[25] Zob. Marian Walczak, *Media elektroniczne w edukacji bibliotekarzy i pracowników informacji naukowej*, Biuletyn Głównej Biblioteki Lekarskiej, 1999, nr 359, s. 67-76; Jennifer Durran, *Developments In Elctronic Image Databases For Art History*, Stanford University Libraries: CoCL (Conservavtion On-Line), 2003. Tryb dostępu: http://palimpsest.stanford.edu/byauth/durran/developments.html.

[26] Remigiusz Sapa, *Zastosowanie benchmarchingu do kształtowania polityki marketingowej serwisu WWW biblioteki akademickiej*, W: *Elektroniczne publikacje w bibliotekach*, red. nauk. Maria Kocójowa, Kraków: Wydaw. UJ, 2002, s. 52-59.

MEDIEVAL MANUSCRIPTS IN THE WORLD INTERNET RESOURCES

Summary

Internet resources for medieval manuscripts are very wide. Apart from single ilustrations coming form well-known manuscripts the internet resources offer access to whole libraries of digitalised manuscripts, their critical editions and scientific apparatus as well as bibliographic databases. The most significant role is played by the libraries, mainly national ones and scientific institutions. However there are also plenty of individual web sites of particular interest created by enthusiasts. The websites recipients are mainly persons profesionally interested in manuscripts: historians of art, bibliologists, medievists as well as students and rare book lovers.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Michał Franczak, Krzysztof Juszczyszyn

Instytut Sterowania i Techniki Systemów, Politechnika Wrocławska

WYKORZYSTANIE JĘZYKA RDF DO OPISU I PRZESZUKIWANIA ZBIORU TEKSTÓW LITERACKICH

Streszczenie

W artykule zaproponowano, bazujący na języku RDF, webowy standard opisu tekstów literackich. Opisano prototypową implementację serwisu wyszukiwawczego umożliwiającego odszukanie tekstów opisanych przy użyciu proponowanego standardu.

1. WSTĘP

Mimo intensywnego rozwoju technik i webowych standardów opisu informacji, wyszukanie w sieci żądanych zasobów wciąż pozostaje nietrywialnym i nierozwiązanym do końca problemem. Wyszukiwarki pełnotekstowe sprawdzają się w przypadku większości stron, jednak jeśli chcemy znaleźć na przykład tekst literacki (a ogólniej: zasób, którego charakterystyka wyszukiwawcza jest zdefiniowana inaczej niż tylko za pomocą zbioru słów kluczowych), wyniki zwracane przez większość serwisów wyszukujących są niezadowalające. Czasem szukamy na przykład czegoś podobnego do ostatnio przeczytanej książki, lub tekstu o wskazanej treści, należącego do określonego gatunku literackiego. Serwisy literackie oferują często bogate opisy swoich zbiorów, ale wyszukiwanie wśród tych zbiorów nadal odbywa się pełnotekstowo. Wciąż mało jest standardów, które pozwoliłyby opisywać *treść* i *rodzaj* informacji, a nie ma już w ogóle standardów pozwalających w szeroki sposób opisać tekst literacki. W dalszej części artykułu przedstawione zostaną założenia i działanie prototypowego serwisu wyszukiwawczego dla tekstów literackich, korzystającego z opartego na RDF standardu opisu zasobów tego typu.

2. STANDARD RDF

Automatyczne przetwarzanie informacji w sieci WWW wiąże się z wieloma problemami natury praktycznej i teoretycznej. Obecnie prowadzone są prace zmierzające do wzbogacenia stron WWW o informację opisującą treść (semantyczną) co prowadzić ma do powstania tzw. sieci semantycznej (Semantic Web).

Język RDF (*Resource Description Framework* [3,10]), wprowadza prosty model danych dla wyrażeń, zapisując je w postaci uporządkowanych trójek postaci {podmiot,

predykat, obiekt} [6]. Pierwszą składową trójki jest podmiot. Drugą jest nazwa własności, dla uproszczenia nazywana dalej własnością. W przypadku podmiotu „człowiek" możemy wskazać przykładowe własności takie jak: wiek, kolor oczu, imię, waga, wzrost itp. Trzecią i ostatnią składową jest wartość. Przykładowa trójka będzie zatem wyglądać następująco: „Chłopiec (podmiot) o imieniu (własność) Piotr (wartość)" lub, krócej, {*Chłopiec, imię, Piotr*}. W przypadku zastosowania RDF do opisu zasobów WWW trójka wygląda następująco: podmiot nazywany jest zasobem (ang. resource) i opisywany jest za pomocą URI (Uniform Resource Identifiers). Własność nazywana jest predykatem (ang. predicate) i może być atrybutem, relacją lub charakterystyką. Trzecim elementem trójki RDF jest obiekt (ang. object), jest on ekwiwalentem wartości (wprowadzenie do składni RDF znaleźć można w [6] oraz na stronach W3C – *www.w3c.org*). Dla przykładu, niech zasobem będzie opowiadanie w serwisie czytaj.org (http://czytaj.org). Tworzymy zdanie: *Opowiadanie zamieszczone pod adresem http://www.czytaj.org/index.php?action=czytaj &id=2&aut=5 nosi tytuł „Mrówczy sens";*

{*http://www.czytaj.org/index.php?action=czytaj&id=2&aut=5, tytuł, „Mrówczy sens"*}

Do chwili obecnej opracowano kilka narzędzi wspomagających zadawanie pytań o dane opisane w RDF. W przypadku danych zapisanych w języku XML mamy do czynienia z hierarchiczną bazą danych, jednak kiedy stosujemy RDF całość struktury skłania się raczej w kierunku bazy relacyjnej. Naturalnym jest zatem, że poszukiwania języka zadawania pytań bazowały na wykorzystaniu już istniejącego języka SQL. Bardzo podobna zatem do SQL jest składnia języków takich jak SquishQL, RDQL, RQL i innych.

Pierwszym, najbardziej intuicyjnym, krokiem w poszukiwaniu rozwiązania, stało się umieszczenie danych RDF w relacyjnej bazie danych. Dane umieszczone w tabelach nie różnią się zbytnio od standardowego trypletu RDF, natomiast do zadawania pytań można użyć języka SQL (mówimy wtedy o jego odmianie dostosowanej do RDF – RDQL) [4]. Jednak dane umieszczone w takich bazach mogą przybrać dość duże rozmiary co wpływa negatywnie na efektywność takiego rozwiązania.

Bardzo przydatnym i ciekawym narzędziem wspomagającym pracę z RDQL jest pakiet class_rdql_db. Pakiet składa się z dwóch klas. Jedna z nich umieszcza dane RDF w bazie danych natomiast druga realizuje zapytania RDQL. Cały pakiet jest udostępniany na licencji GPL i dostępny bez opłat w serwisie SourceForge. Pakiet calss_rdql_db jest rozwiązaniem, które zostało wybrane dla potrzeb omawianego zagadnienia.

3. OPIS TEKSTU LITERACKIEGO

Dość ważnym elementem przy opisie danych za pomocą RDF jest schemat opisu. Słowniki RDF nie są niczym innym jak przestrzeniami nazw zgodnymi z RDF Schema. Jednym z takich podstawowych słowników jest Dublin Core [2]. Schemat DC służy do opisu zasobów przechowywanych w sieci. Innym przykładem może być Postcon, schemat stworzony przez Shelley Powers [6] do porządkowania wpisów w dziennikach sieciowych (ang. *Web Log*, skr. *Blog*).

3.1. Dublin Core

Standard DC opisuje bibliograficznie zasób za pomocą 15 podstawowych elementów. Każda definicja DC dotyczy opisywanego zasobu. Dublin Core opisuje metadane za pomocą 15 podstawowych elementów. Każdy z elementów Dc jest zdefiniowany za pomocą 10 atrybutów [2][5]. Za pomocą 15 zaprezentowanych elementów można opisać

bibliograficznie dowolny zasób, który można jednoznacznie zidentyfikować. Spróbujmy zatem opisać przykładową stronę www:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/metadata/dublin_core#">
        <rdf:Description about="http://czytaj.org">
          <dc:Title>Czytaj.org</dc:Title>
          <dc:Description>Serwis artystyczno-literacki </dc:Description>
          <dc:Publisher>Serwis Czytaj.org</dc:Publisher>
          <dc:Contributor>Grupa stricte.net</dc:Contrinutor>
          <dc:Date>2001-02-20</dc:Date>
        <dc:Format>text/html</dc:Format>
        <dc:Language>pl</dc:Language>
</rdf:Description></rdf:RDF>
```

Jak widać z powyższego przykładu, DC oferuje już znacznie bogatsze możliwości opisu tekstu literackiego niż metainformacje jakie możemy umieścić w kodzie HTML. To jednak jeszcze nie wystarcza do opisania tekstu literackiego.

3.2. Proponowany schemat opisu tekstu literackiego

Istnieje wiele metod klasyfikacji literatury. Jeśli chodzi o sam opis ogólny tekstu, mamy nawet wiele standardów. Kiedy jednak chcemy opisać jego treść, ustalonych standardów brakuje. W prototypowym serwisie wyszukiwawczym zaproponowano wykorzystanie trzech metod: ogólnej (bibliograficznej), opartej na morfologii bajki Włodzimierza Proppa oraz opisu mitologicznego.

3.2.1. Metoda ogólna

Podstawą opisu bibliograficznego w wersji zaproponowanej w omawianym rozwiązaniu sa składowe (wystepujące jako elementy dokumentu RDF) takie jak: *autor, tytuł, gatunek, właściciel_praw, wydanie,...*

Słownik ogólny jest najbardziej rozbudowanym słownikiem całego standardu. Pozwala on zdefiniować podstawowe informacje o tekście. Zawiera on 19 elementów które możemy podzielić na dwie grupy. Pierwsza grupa to elementy dotyczące samego tekstu druga to elementy opisujące publikacje tekstu. W tych 19 elementach można już zawrzeć ogólne informacje na temat opisywanego tekstu i w tym sensie ich rola jest identyczna z Dublin Core, aczkolwiek funkcjonalność proponowanego słownika jest nieco szersza (dostosowano ją do opisu zasobów WWW).

Implementacja powyższych elementów, jako przestrzeń nazw, znajduje się pod adresem *http://stricte.net/litdesc/elements/0.1/general/#*. Całość standardu jest zgodna z RDF Schema. Wszystkie elementy prócz definicji jako własności (properties) zostały przypisane do poszczególnych klas.

3.2.2. Metoda oparta na morfologii bajki według W. Proppa

Przedstawiona w poprzednim rozdziale metoda dotyczyła tylko opisu bibliograficznego tekstu. Pamiętamy jednak, że głównym celem jest możliwość wyszukiwania tekstu na podstawie jego treści. Wielu teoretyków literatury próbowało opisać w uniwersalny sposób treść dzieła prozaicznego czy też lirycznego. O ile w przypadku liryki większość prób spełzła na niczym o tyle w przypadku epiki cele po części zostały osiągnięte. Wielką rolę w tej dziedzinie odegrali formaliści i strukturaliści rosyjscy a wśród nich Włodzimierz Propp.

Propp przeprowadził interfabularne porównanie bajek rosyjskich. Użył do tego około czterystu pięćdziesięciu tradycyjnych bajek. Udało mu się opisać ogólną budowę każdej bajki za pomocą uniwersalnych elementów. Najważniejszymi i najlepiej opisującymi treść bajki są funkcje działających postaci. Propp wydzielił 31 podstawowych funkcji i podał kolejność, w jakiej pojawiają się w tekście literackim. Występowanie każdej z funkcji w bajce nie jest obligatoryjne, lecz kolejność ich występowania jest z góry ustalona. Poniżej prezentowane są funkcje tworzące podstawę morfologiczną bajek magicznych. Bajka rozpoczyna się jakąś sytuacją wyjściową. Sytuacja ta nie jest traktowana jak funkcja. Po sytuacji początkowej pojawiają się następujące funkcje (ze względu na rozmiar listy pominięto większość funkcji) [8]:

I. Jeden z członków rodziny opuszcza dom (odejście)
II. Bohaterowi zostaje wydany zakaz
III. Zakaz zostaje naruszony (naruszenie)
(...)
XXX. Przeciwnikowi zostaje wymierzona kara
XXXI. Wesele

Na potrzeby serwisu wyszukiwawczego wyróżniono trzy podstawowe elementy dla drugiej metody opisu tekstu, przy czym pierwszy element jest rozbudowany i ma zamknięty słownik wartości (odpowiadających zbiorowi 31 funkcji wyodrębnionych przez Proppa). Drugi z tych elementów (typ bajki) ma trzywartościowy zamknięty słownik, trzeci zaś jest rozbudowanym elementem zawierającym opisy narracyjne. Drzewo elementów prezentuje się następująco:

- funkcja
- typ bajki
- atrybuty postaci (wygląd - osobliwości związane z postacią - miejsce w którym przebywa)

Z punktu widzenia standardu opisu, najlepszym rozwiązaniem w przypadku atrybutów postaci jest przytoczenie fragmentów bajki, które dotyczą poszczególnych zbiorów i zamknięcie ich specjalnymi znacznikami. Wyszukiwanie w tych fragmentach odbywa się pełnotekstowo. Elementy nie są co prawda obligatoryjne, jednak należy pamiętać, że opis bohatera czy też typ bajki u Proppa mają raczej charakter pomocniczy, a najważniejszą częścią opisu jest występowanie funkcji. Nie jest zatem dobrym pomysłem użycie do opisu tylko elementów opisujących bohatera.

Zaproponowana metoda opisu umożliwia implementację serwisu wyszukiwawczego o specyficznych cechach użytkowych. Teoretyk literatury otrzyma w tym przypadku narzędzie, które pozwoli mu wyszukiwać do badań bajki spełniające odpowiednie wymogi. Zwykły użytkownik dzięki takiemu opisowi w prosty sposób znajdzie bajkę podobną (pod względem fabuły, występujących postaci etc...) do tej, którą przed chwilą przeczytał i która mu się spodobała.

3.2.3. Metoda mitograficzna

Trzecia metoda opisu tekstu literackiego została wywiedziona z krytyki mitograficznej. Krytyka mitograficzna oparta jest na badaniach C.G. Junga i wcześniej Z. Freuda i zajmuje się wyszukiwaniem odpowiednio zdefiniowanych motywów mitycznych w twórczości. W naszym przypadku będzie to twórczość prozatorska, lecz można stosować ten rodzaj krytyki również do innych dziedzin sztuki (np. malarstwa).

Z punktu widzenia omawianego projektu ważne jest, że w każdym tekście krytyk może wyodrębnić poszczególne motywy mityczne. Dzięki temu możliwe jest swoiste opisanie treści utworu narracyjnego. W odróżnieniu od metody opartej na morfologii bajki Proppa, tutaj mamy zdecydowanie szersze pole działania. Teksty nie muszą spełniać warunków jakiejś szczególnej poetyki, zatem opisać tym sposobem można o wiele więcej tekstów literackich. Jeśli jednak w przypadku Proppa, mogliśmy powiedzieć, że teksty w których występowały te same funkcje bohatera były do siebie podobne o tyle w przypadku motywów mitycznych tak sformułowanych wniosków wysunąć nie można.

Ważnym elementem w przypadku tej krytyki jest pochodzenie kulturowe tekstu. Raczej nie stosuje się szukania mitów orientalnych w tekstach pochodzących z kultury judeo-chrześcijańskiej, której motywy mityczne najczęściej pochodzą z mitologii greków [9]. Metoda mitograficzna korzysta z następujących elementów:

Element: Krąg kulturowy
Identyfikator: culture
Komentarz: Zawiera informację o kulturowym pochodzeniu tekstu.

Element: Motyw
Identyfikator: motif
Komentarz: Zawiera nazwę występującego motywu. Nazwa powinna być krótka i powinna być znaną ogólnie nazwą tego motywu. W przypadku pokrywania motywów z dwóch źródeł występujących w jednym kręgu kulturowym wybieramy starsze źródło.

Słownik mitograficzny to krótka przestrzeń nazw, składa się tylko z dwóch elementów, jednakże element *Motyw* może powtarzać się wielokrotnie i w zależności od długości tekstu może powtórzyć się nawet kilkadziesiąt razy. Powyższy słownik wydaje się najprostszym ze wszystkich zaprezentowanych, jednak to właśnie on daje największe możliwości opisania treści. W przypadku nazw motywów ogólnie znanych, jesteśmy w stanie przybliżyć sobie treść tekstu. Również w przypadku wyszukiwania tekstu, szukanie według motywów mitycznych najtrafniej ze wszystkich metod określa treść tekstu.

4. IMPLEMENTACJA

Warto zwrócić uwagę na fakt, że w odróżnieniu na przykład od DublinCore, całość standardu jest rozbita na mniejsze słowniki zamiast jednego dużego. Pozwala to na sprawniejszy wybór metody opisu i znacznie ułatwia w przyszłości dodawanie własnych słowników do standardu. Implementacja każdego ze słowników została umieszczona pod różnymi adresami, dzięki temu odwołując się do przestrzeni nazw możemy odwołać się tylko do wybranych słowników. Wszystkie słowniki zostały umieszczone pod adresem *http://stricte.net/litdesc/elements/0.1/skrót_nazwy_słownika/#*. Skrót „*litdesc*" pochodzi od roboczej nazwy standardu Literature Description Schema Project. Wersja robocza standardu otrzymała numer 0.1. Prócz poszczególnych słowników został stworzony także słownik ogólny, znajdujący się pod adresem *http://stricte.net/litdesc/elements/0.1/#*. W słowniku tym zdefiniowana jest klasa główna określająca zasób, oraz dwie klasy i jedna własność pomocnicza. Poniżej zamieszczono definicję słownika podstawowego:

```
<rdfs:Class rdf:about="http://stricte.net/litdesc/elements/0.1/#Resoucre">
        <rdfs:isDefinedBy rdf:resource="http://stricte.net/litdesc/elements/0.1/" />
        <rdfs:label>Resource</rdfs:label>
        <rdfs:comment>Definicja tekstu literackiego jako zasobu /rdfs:comment>
        <rdfs:subClassOf rdfs:resource="http://www.w3.org/TR/WD-rdf-schema#Resource" />
</rdfs:Class>
<rdfs:Property rdf:about="http://stricte.net/litdesc/elements/0.1#required">
        <rdfs:isDefinedBy rdf:resource="http://elvis.eti.pg.gda.pl/schema#" />
        <rdfs:label>required</rdfs:label>
        <rdfs:range rdfs:resource="http://stricte.net/litdesc/elements/0.1/#Boolean" />
</rdfs:Property>
```

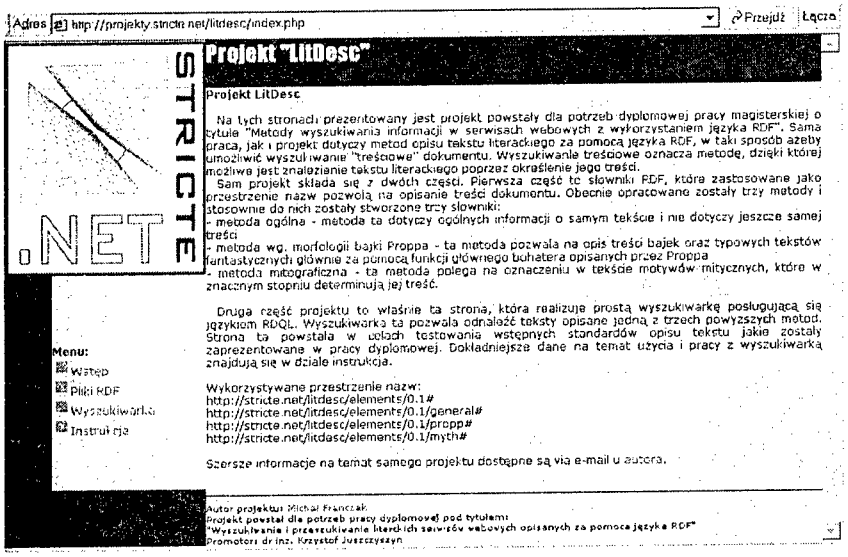
Na potrzeby pierwszych eksperymentów zostały wybrane cztery teksty literackie pochodzące z serwisu literacko-artystycznego czytaj.org. Wszystkie teksty zostały opisane metodą ogólną, następnie dwa z nich zostały opisane metodą opartą na morfologii bajki, a kolejne dwa metodą mitograficzną. Opisów na poziomie teoretyczno-literackim dokonał mgr Artur Telwikas - teoretyk literatury. Opisane zostały następujące teksty:

1. „*Córka przewoźnika*" autor: Jagoda Przybyłowska – metoda Proppa
2. „*Baldur*" autor: Emilian Semczak – metoda Proppa
3. „*Torment*" autor: Emilian Semczak – metoda mitograficzna
4. „*Mrówczy sens*" autor: Michał Franczak – metoda mitograficzna

Pliki zawierające opisy powyższych tekstów dostępne są pod adresami:

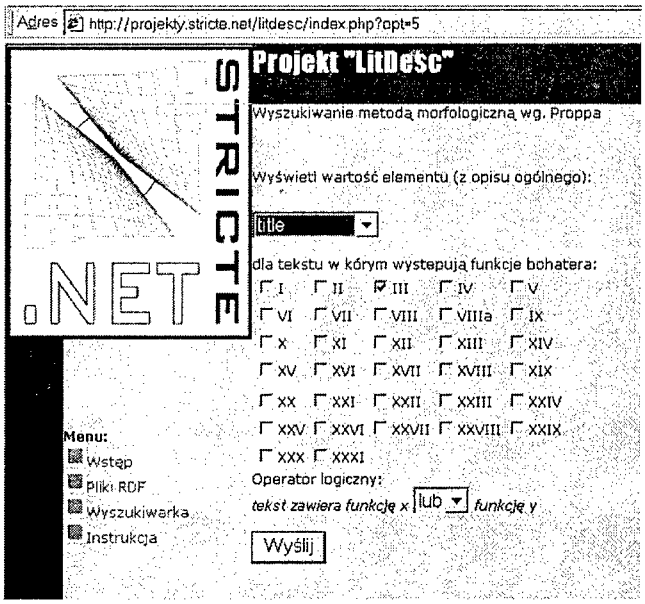
http://projekty.stricte.net/litdesc/?.rdf (gdzie „?" = 1,2,3,4 – zgodnie z listą powyżej)

Prototyp wyszukiwarki pozwalającej odnajdywać teksty opisane za pomocą zaproponowanego standardu został umieszczony pod adresem *http://projekty.stricte.net/litdesc/*.



Rys. 1. Strona główna serwisu

Rozwiązanie wykonano w technologiach PHP oraz JavaScript (interfejs strony). Oprogramowanie działa w oparciu o serwer Apache z modułem PHP, serwer baz danych „MySQL" oraz pakiet PHP „class_rdql_db". Pakiet ten jest rozpowszechniany jest na podstawie licencji GNU i dostępny pod adresem *http://phpxmlclasses.sourceforge.net* [1]. W przypadku drugiej metody wyszukiwania musimy wybrać, których funkcji bohatera poszukujemy w tekście. Mamy też możliwość wyboru operatora logicznego w przypadku kilku funkcji. Na wyjściu możemy otrzymać jeden z elementów opisu ogólnego. Przykładowo (rys. 2):



Rys. 2. Interfejs wyszukiwania dla opisu morfologicznego wg. Proppa

Pytanie: *Pokaż tytuły tekstów w których bohater łamie nadany mu zakaz.*

Realizacja: *Wyjście ustawione na title, zaznaczamy checkbox przy funkcji III ponieważ to właśnie ona dotyczy złamania zakaz.*

Wyświetlany wynik:

Wynik wyszukiwania: *Córka Przewoźnika III*

Odpowiednie zapytanie RDQL:

SELECT ?title, ?y FROM * WHERE (?x, <ldpropp:function>, ?y),
(?x,<ldgeneral:title>,?title) AND ?y=="III" USING rdf for <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, ld for <http://stricte.net/litdesc/elements/0.1#>, ldpropp for <http://stricte.net/litdesc/elements/0.1/propp#>, ldgeneral for <http://stricte.net/litdesc/elements/0.1/general#>

5. PODSUMOWANIE

Zaproponowane rozwiązanie wymaga jeszcze dużego nakładu pracy i z pewnością nie byłoby możliwe, gdyby nie połączenie dwóch jakże odległych dziedzin nauki – teorii literatury i informatyki. Uzyskane wyniki, pokazują, że udało się znaleźć drogę do treściowego wyszukiwania tekstu literackiego. Zaproponowano standard opisu, następnie zaimplementowano prototypowy serwis i wyszukiwarkę. W przyszłości będzie ona udoskonalana a przede wszystkim uczyniona dużo prostszą w obsłudze, dzięki czemu każdy będzie mógł bez problemu z niej korzystać.

Znaczący efekt praktyczny może być osiągnięty przez łączne zastosowanie kilku metod wyszykiwania tekstów literackich (opartych np. na opisie bibliograficznym i mitograficznym). Jednocześnie standard opisu powinien być w miarę możliwości otwarty ze względu na definiowanie nowych metod opisu oraz wybór wyszukiwania według żądanej metody. Stworzona na potrzeby eksperymentu wyszukiwarka znakomicie nadaje się dla serwisu prezentującego teksty literackie, jednak bardziej ambitnym celem byłoby stworzenie wyszukiwarki globalnej. Wyszukiwarka działaby na podobnej zasadzie jak Google, z tą jednak różnicą, ze zamiast całych stron zbierałaby do bazy tylko pliki RDF zgodne ze standardem. Takie rozwiązanie pozwoliłoby na przeszukiwanie całego Internetu w poszukiwaniu tekstu literackiego o żądanej treści. Pliki RDF zbierałby robot sieciowy, następnie umieszczając w bazie. Rozwiązanie to jest o tyle lepsze od tego stosowanego w Google, że w bazie znajdowałyby się metainformacje opisowe nie zaś cały tekst. Pozwoliłoby to znacznie odciążyć bazy i zmniejszyć ich rozmiary.

BIBLIOGRAFIA

[1] Argerich L., http://phpxmlclasses.sourceforge.net, 2003.
[2] http://DublinCore.org
[3] Lassila O. *Resource Description Framework (RDF) Model and Syntax Specification* http://www.w3.org/TR/REC-rdf-syntax/, 1999.
[4] Melnik S., *Storing RDF in a relational database*, http://www-db.stanford.edu/~melnik/rdf/db.html, 2001.
[5] Nahotko M., *Dublin Core Metadata Element Set*, Version 1.1: Reference Description – przekład polski, 2000.
[6] Powers S. *Practical RDF*, O'Reilly & Associates, Inc. Sebastopol (CA) lipiec 2003
[7] Propp W. *Nie tylko bajka* tłum. Danuta Ulicka, PWN, Warszawa 2000,
[8] Siba K. *Władimir Propp: Morfologia bajki.*
[9] Tarnogórska M.J. *Mit jako kategoria w badaniach literaturoznawczych* http://www.hamlet.filo.pl/kont/mitkat.html, 2002.
[10] *RDF Model Theory*, W3C Working Draft (www.w3c.org), 14 February, 2002.

RDF-BASED METHOD FOR LITERARY TEXT DESCRIPTION AND SEARCH

Summary

The paper deals with description and searching of literary texts over Internet. RDF-based standard for literary text description was proposed and a search engine based on proposed solution was implemented.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Teresa Grabowska, Michał Zawadzki

Katedra Inżynierii Oprogramowania, Politechnika Gdańska

ZASTOSOWANIE RELACYJNYCH BAZ DANYCH I LOGIKI OPISOWEJ W SYSTEMACH INTEGRACJI DANYCH

Streszczenie

Integracja różnych źródeł danych jest aktualnie jednym z fundamentalnych problemów związanych z zarządzaniem wiedzą. Bieżący stan rozwoju systemów zarządzania wiedzą wskazuje, że jego bardzo obiecującym praktycznym rozwiązaniem jest wykorzystanie modelu relacyjnego do zapisu semantycznego baz wiedzy. Artykuł przedstawia najbardziej popularne rozwiązania problemu integracji danych: podejście standardowe, podejście rozszerzone o wiedzę dziedzinową oraz podejście hybrydowe. W artykule opisane zostały wady i zalety poszczególnych rozwiązań oraz istniejące na rynku systemy integracji danych, wraz z analizą ich użyteczności w rzeczywistych warunkach biznesowych. Celem artykułu jest analiza tych rozwiązań pod kątem ich wykorzystania do zapisu pojęć semantycznych z baz wiedzy w modelu relacyjnym.

1. WSTĘP

Celem systemów integracji danych jest dostarczenie jednolitego interfejsu do wielu, dostępnych w sieci, heterogenicznych źródeł informacji. Oznacza to, że systemy integracji danych mają umożliwiać użytkownikowi zadanie zapytania wyrażającego jego potrzebę, uwalniając go od poszukiwania odpowiedzi. Stąd wyszukiwanie odpowiednich źródeł danych, w których może być zawarta częściowa lub pełna odpowiedź na zadane pytanie, interakcja z każdym z tych źródeł z osobna i scalanie uzyskanych informacji w spójną odpowiedź są zadaniami systemu integracji danych, nie zaś jego użytkownika. Problem integracji danych jest zagadnieniem złożonym, co spowodowane jest:

- powiązaniem i nakładaniem się danych zapisanych w różnych źródłach,
- niejednolitymi modelami i schematami danych oraz
- różnorodnymi możliwościami przetwarzania zapytań przez źródła danych [1].

Rozwiązanie problemu integracji danych wymaga zdefiniowania i opracowania dwóch zagadnień:

- sposobu definiowania opisu zawartości różnorodnych źródeł danych,
- procesu uzyskiwania odpowiedzi na zapytania wysłane do systemu integracji danych, na podstawie wcześniej zdefiniowanego opisu źródeł [2].

Bieżący stan rozwoju systemów zarządzania wiedzą wskazuje, że serwery relacyjne dostarczają najbardziej efektywnego sposobu przetwarzania danych. Nie dostarczają one

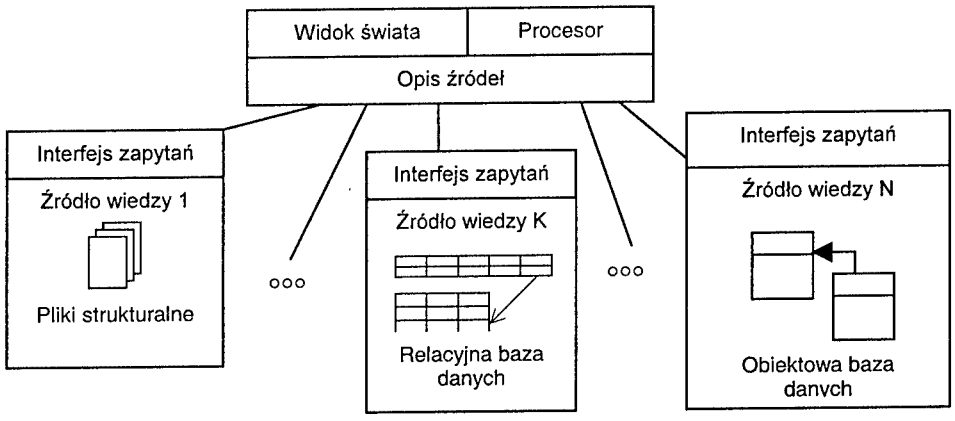
jednak zazwyczaj możliwości semantycznego opisu zawartości różnorodnych źródeł danych. Tę własność posiadają jednak systemy ukierunkowane na semantyczną analizę danych z wykorzystaniem ontologii. Dlatego też powiązanie ze sobą obu tych rozwiązań umożliwia stworzenie praktycznego systemu integracji danych. Przy takim podejściu systemy opisujące semantykę danych zajmują się zagadnieniem definiowania opisu zawartości różnorodnych źródeł danych, zaś serwery relacyjne procesem uzyskiwania odpowiedzi na zapytanie.

Systemy integracji danych możemy podzielić na dwie grupy pod względem dwóch przyjętych dla tych systemów architektur. Pierwsza z nich jest oparta na globalnym schemacie opisującym wiedzę zawartą w heterogenicznych źródłach danych; w tej architekturze użytkownicy postrzegają system integracji danych jako centralny system informacji. Druga architektura unika konstruowania globalnego schematu na rzecz stosowania różnorodnych narzędzi wymiany informacji [3]. Rozwiązanie łączące serwery relacyjne z logiką opisową wymaga architektury opartej na globalnym schemacie wiedzy, dlatego też systemy o takiej architekturze są analizowane w tym artykule.

Celem artykułu jest przedstawienie istniejących rozwiązań problemu integracji danych wykorzystujących model relacyjny lub logikę opisową, będącą jednym z najbardziej popularnych formalizmów semantycznego zapisu wiedzy [2]. Przedstawione zostały dostępne na rynku systemy, ich wady i zalety, a przede wszystkim dokonano analizy ich praktycznego zastosowania.

2. ARCHITEKTURA SYSTEMÓW INTEGRACJI DANYCH OPARTYCH NA GLOBALNYM SCHEMACIE WIEDZY

Architektura systemów integracji danych opartych na globalnym schemacie wiedzy została przedstawiona na rys. 1. Systemy o tej architekturze przedstawiają użytkownikowi całościowy obraz pewnej dziedziny wiedzy, określany jako *widok świata*. Użytkownik może zadać zapytanie w terminach określonych przez ten widok, niezależne od źródeł danych, ich rozmieszczenia lub zdefiniowanej dla nich metody dostępu. Należy zauważyć, że widok świata jest jedynie schematem wiedzy. Dane potrzebne do tego, aby odpowiedzieć na zadane przez użytkownika zapytanie, znajdują się w heterogenicznych *źródłach wiedzy*.



Rys. 1. Architektura systemu integracji danych opartego na globalnym schemacie wiedzy

W celu znalezienia odpowiedzi na zapytanie i wykonania tego zadania wydajnie, *procesor zapytań* wykorzystuje *opisy źródeł*. Opisy źródeł zazwyczaj obejmują takie informacje jak:

- związek pomiędzy zawartością źródła danych a widokiem świata,
- zależności pomiędzy źródłami danych,
- informacja czy dane zawarte w źródle danych w pełni opisują wiedzę dotyczącą fragmentu widoku świata,
- rodzaje zapytań, na jakie źródło danych może odpowiedzieć.

Dla każdego źródła danych jest zdefiniowany *interfejs zapytań* wykonujący zrozumiałe dla niego zapytanie, przetworzone wcześniej przez procesor zapytań [4].

3. KLASYFIKACJA SYSTEMÓW INTEGRACJI DANYCH OPARTYCH NA GLOBALNYM SCHEMACIE WIEDZY

Systemy integracji danych oparte na globalnym schemacie wiedzy można podzielić na trzy podstawowe grupy bazujące na: podejściu standardowym, podejściu rozszerzonym o model dziedzinowy i podejściu hybrydowym. Podejście standardowe zakłada, że zarówno model globalnego schematu, jak i lokalne schematy źródeł wiedzy są relacyjne. Jednak w wielu zastosowaniach potrzebny jest bogatszy model dziedzinowy. Dlatego podejście rozszerzone o wiedzę dziedzinową zakłada użycie bogatszego globalnego schematu wiedzy opartego na logice opisowej. Podejście hybrydowe dodatkowo stosuje reguły Horna [1], jako że logika opisowa ma również swoje ograniczenia, zwłaszcza w zakresie wyrażania relacji pomiędzy danymi.

3.1. Podejście standardowe

Podejście standardowe nakłada określone założenia na poszczególne komponenty architektury systemu. Przede wszystkim widok świata musi być opisany za pomocą modelu relacyjnego. Jest on postrzegany jako zbiór wirtualnych relacji, nigdzie bezpośrednio nie przechowywanych. Źródła wiedzy są również opisywane w modelu relacyjnym. Schematy źródeł wiedzy nazywamy lokalnymi schematami w odróżnieniu od globalnego schematu wiedzy. Opis źródeł definiuje związki pomiędzy relacjami w globalnym schemacie a relacjami w lokalnych schematach. Najbardziej popularne są dwa podejścia: Global As View (GAV) oraz Local As View (LAV). Podejście Global As View polega na tym, że dla każdej wirtualnej relacji R w globalnym schemacie definiujemy zapytanie operujące na relacjach zdefiniowanych w lokalnych schematach, specyfikując, w jaki sposób uzyskać krotki relacji R ze źródeł wiedzy. Z kolei podejście Local As View polega na tym, że opis źródeł jest podawany w przeciwnym kierunku, co oznacza, że zawartość źródeł wiedzy jest opisana w postaci zapytań dla globalnego schematu.

Zadanie zapytania w systemach integracji danych bazujących na podejściu standardowym polega na przekształceniu zapytania zadanego w terminach globalnego schematu w zbiór zapytań, które odnoszą się w sposób bezpośredni do źródeł wiedzy. Proces ten bezpośrednio korzysta z opisów źródeł, wykorzystując zdefiniowane zależności pomiędzy relacjami. Następnie zapytanie jest optymalizowane i wykonywane bezpośrednio na źródłach wiedzy. Dane zwracane przez źródła wiedzy są przetwarzane do formatu zrozumiałego przez system integracji danych i scalane [1].

3.2. Podejście rozszerzone o wiedzę dziedzinową

Globalnym schematem wiedzy dla systemów zbudowanych zgodnie z podejściem rozszerzonym o wiedzę dziedzinową jest ontologia, która opisuje zawartość źródeł wiedzy dostępnych w pewnej dziedzinie. Ontologia składa się z konceptów i relacji pomiędzy nimi. Jednym z najbardziej popularnych formalizmów jej zapisu jest logika opisowa. Dlatego też jest ona chętnie wykorzystywana do modelowania wiedzy dziedzinowej w systemach integracji danych. Analogicznie do globalnego schematu modelowana jest zawartość źródeł wiedzy. Następnie definiowane są związki pomiędzy konceptami i relacjami w schemacie globalnym a konceptami i relacjami w lokalnych schematach źródeł wiedzy. Tak jak w przypadku podejścia standardowego, zapytanie zadane w terminach globalnego schematu jest przekształcane do zbioru zapytań zrozumiałych dla poszczególnych źródeł wiedzy. Dla systemów bazujących na podejściu rozszerzonym o wiedzę dziedzinową nie jest zdefiniowany konkretny sposób przetwarzania zapytań. Różne systemy podchodzą do tego zagadnienia inaczej. Jedno z podejść zastosowane w systemie SIMS [5] korzysta z operatorów realizujących: przekształcanie konceptów z poziomu dziedzinowego do poziomu źródeł wiedzy, zapisywanie konceptów w terminach konceptów bardziej ogólnych, zapisywanie konceptów w postaci konceptów bardziej wyspecjalizowanych i zapisywanie konceptów poprzez zbiór innych konceptów zawierających wszystkie instancje danego konceptu [6]. Inne podejście zostało zastosowane w systemie PICSEL. Integruje ono metody opisu źródeł zdefiniowane dla podejścia standardowego, tj. Global As View i Local As View, w taki sposób, aby złagodzić ich wady [7].

Opracowano również model schematu wiedzy opisujący zależności pomiędzy schematami źródeł wiedzy. W odróżnieniu od wcześniej przedstawionych rozwiązań model ten nie definiuje bezpośrednio konceptów i relacji pomiędzy nimi, a zawiera jedynie twierdzenia opisujące związki pomiędzy konceptami zdefiniowanymi w różnych schematach źródeł wiedzy [3].

3.3. Podejście hybrydowe

Systemy hybrydowe są specjalną klasą systemów reprezentacji wiedzy, które składają się z dwóch lub więcej podsystemów dotyczących różnych części bazy wiedzy i specyficznych dla nich procedur wnioskowania. Łączą wiedzę i procedury wnioskowania różnych podsystemów w celu znalezienia odpowiedzi na zapytanie zadane przez użytkownika [8]. Podejście hybrydowe zastosowane w systemach integracji danych łączy dwa rozwiązania: reguły Horna i logikę opisową. Reguły Horna są atrakcyjne, ponieważ są dla nich opracowane praktyczne i wydajne procedury wnioskowania. Jednak reguły Horna nie są wystarczającym narzędziem modelowania dziedziny wiedzy i jej hierarchicznej struktury. Funkcję tę w systemach hybrydowych spełnia logika opisowa [9].

3.4. Porównanie rozwiązań stosowanych w systemach integracji danych

Podejście standardowe charakteryzuje się ograniczonym modelem dziedzinowym, uniemożliwiającym budowanie bogatego, semantycznego modelu danych. Dodatkowo proces tworzenia opisów źródeł w momencie zmiany, usunięcia lub dodania nowego źródła wiedzy musi być ponownie powtarzany (GAV), bądź też proces przetwarzania zapytań jest trudny obliczeniowo i koncepcyjnie (LAV). Model standardowy jest odpowiedni dla źródeł wiedzy będących relacyjnymi bazami danych. W przypadku dowolnego innego modelu źródeł danych podejście to jest trudne do praktycznego zastosowania.

Podejście rozszerzone o wiedzę dziedzinową pokonuje dwie bardzo ważne wady podejścia standardowego. Przede wszystkim logika opisowa umożliwia tworzenie bogatej semantycznie struktury hierarchicznej opisującej wiedzę dziedzinową. Dodatkowo przetwarzanie zapytań jest wykonywane dynamicznie, a dodanie nowego źródła nie wymaga przebudowywania opisu pozostałych źródeł wiedzy. Takie podejście jest bardziej elastyczne i skalowalne od podejścia standardowego.

Podejście hybrydowe w porównaniu z podejściem rozszerzonym o wiedzę dziedzinową oferuje większe możliwości wnioskowania, dzięki językowi zapytań rozszerzonemu o zapytania, konstruowane jako koniunkcja wyrażeń.

Analizując powyższe rozwiązania, można zauważyć, że dokonany podział pokazuje drogę ewolucji systemów integracji danych. Systemy bazujące na podejściu standardowym to systemy chronologicznie najstarsze. Systemy rozszerzone o wiedzę dziedzinową, jak sama nazwa wskazuje, są następcami systemów bazujących na podejściu standardowym. Systemy hybrydowe zaś są jeszcze ciągle w fazie rozwoju.

4. DOSTĘPNE NA RYNKU SYSTEMY INTEGRACJI DANYCH

Systemy dostępne na rynku wykorzystują opisane wcześniej podejścia. W większości systemy te ewoluowały od podejścia standardowego do podejścia rozszerzonego o wiedzę dziedzinową, natomiast niektóre z nich są prekursorami podejścia hybrydowego. Przedstawione systemy są wzorcowymi systemami wykorzystującymi globalny schemat wiedzy.

4.1. Information Manifold

System Information Manifold został opracowany w Laboratoriach Bell AT&T [10]. Udostępnia on zintegrowany, oparty na WWW interfejs zapytań do wielu heterogenicznych źródeł wiedzy. Ze względu na fakt, iż widok świata konstruowany jest przy użyciu logiki opisowej rozszerzonej o klauzule Horna, system ten należy do grupy systemów wykorzystujących podejście hybrydowe.

Information Manifold ukierunkowany jest na źródła wiedzy o ustalonej strukturze. Ważną cechą systemu jest użycie elastycznego mechanizmu opisu zawartości źródeł opartego na podejściu Local As View [10]. Pozwala on na precyzyjne opisanie zawartości źródeł wiedzy, a przez to na rozróżnienie źródeł o podobnej semantycznie zawartości. Dodatkowo pozwala na dodawanie i usuwanie źródeł wiedzy bez potrzeby zmiany opisu pozostałych źródeł.

W systemie zastosowano stosunkowo efektywny algorytm „wiadra" (ang. *bucket algorithm*) [1], w którym procesor zapytań przetwarza zapytania użytkownika na zapytania skierowane bezpośrednio do źródeł danych. Algorytm ten wykorzystuje mechanizm wnioskujący, pozwalający określić, które źródła wiedzy zawierają informacje dotyczące przetwarzanego zapytania. Zastosowanie tego algorytmu pozwala znacznie zredukować liczbę nadmiarowych przekształceń zapytań (odpytywane są tylko źródła związane z zapytaniem) oraz gwarantuje sensowność i kompletność odpowiedzi. Wykorzystane zostały również efektywne algorytmy tworzenia planów wykonania zapytań.

Niestety, mimo zastosowania efektywnych algorytmów, Information Manifold cechuje się nienajlepszą wydajnością. Słabą stroną systemu jest również brak obsługi danych semistrukturalnych, a także wymóg standaryzacji poszczególnych schematów źródeł wiedzy.

System używany jest do przeszukiwania strukturalnych źródeł danych dostępnych w sieci WWW, a także w Laboratoriach Bell AT&T do przeszukiwania zasobów wewnętrznych.

4.2. SIMS – „Services and Information Management for decision Systems"

Celem systemu SIMS [5] opracowanego w Information Sciences Institute (ISI) jest zapewnienie dostępu do heterogenicznych, rozproszonych źródeł wiedzy przechowywanej w postaci danych ustrukturalizowanych i częściowo ustrukturalizowanych. W systemie SIMS tworzony jest model dziedzinowy dla konkretnej dziedziny aplikacji, przy użyciu języka reprezentacji wiedzy LOOM (język modelowania semantycznego) w celu zbudowania słownika opisującego koncepty z dziedziny, właściwości konceptów oraz relacje między nimi. Jest to podejście rozszerzone o wiedzę dziedzinową. Zapytania do systemu zadawane są przy użyciu terminów zdefiniowanych w globalnym schemacie wiedzy. W celu uzyskania odpowiedzi na zapytania procesor zapytań używa operatorów przekształcających zapytanie i wyznaczających zbiór odpowiednich źródeł oraz sposób zintegrowania pozyskanych informacji. Taka konstrukcja pozwoliła na uzyskanie elastycznego i skalowalnego systemu, który potrafi dynamicznie obsługiwać nowo dodawane źródła wiedzy.

Przetwarzanie zapytań odbywa się w czterech krokach: przetworzenie zapytania (transformacja na koncepty źródłowe), planowanie dostępu do zapytań (wyznaczanie planu wykonania), semantyczna optymalizacja planu (na podstawie informacji o zawartości źródeł) oraz jego wykonanie (dynamicznie obsługiwane są błędy źródeł – w razie ich wystąpienia przeprowadzana jest zmiana planu).

System SIMS znalazł zastosowanie między innymi w firmie Boeing, gdzie służy do zarządzania danymi lokalnymi.

4.3. PICSEL

W systemie PICSEL do budowy globalnego schematu wiedzy wykorzystywany jest język CARIN [9], jako podstawowy formalizm logiczny służący do opisywania zarówno dziedziny aplikacji jak i zawartości źródeł wiedzy. Ze względu na użycie logiki opisowej system ten korzysta z podejścia rozszerzonego o wiedzę dziedzinową. Opis zawartości źródeł definiuje się poprzez metodę integrującą podejścia Local As View i Global As View [7]. Procesor zapytań oparty jest na implementacji mechanizmu wnioskującego języka CARIN, dzięki czemu możliwe jest dodawanie nowych źródeł danych przy małym nakładzie pracy – należy przygotować jedynie opis źródła i udostępnić go w globalnym schemacie.

System PICSEL używany był do internetowej obsługi francuskiego biura podróży Degriftour.

5. ZAKOŃCZENIE

Logika opisowa stosowana w systemach integracji danych jako formalizm zapisu modelu dziedzinowego ma liczne zalety. Przede wszystkim umożliwia precyzyjne i niedwuznaczne opisywanie semantyki wraz z możliwością definiowania ograniczeń nakłada-

nych na koncepty i relacje pomiędzy tymi konceptami. Logika opisowa umożliwia również przeprowadzanie wnioskowania w zakresie:

- hierarchii konceptów, ich definicji i relacji pomiędzy nimi (TBox) oraz
- wystąpień konceptów i relacji między nimi (ABox).

Pierwszy zakres obejmuje sprawdzenie:

- wewnętrznej niesprzeczności konceptów – czy koncept może mieć wystąpienia,
- relacji zawierania między konceptami – czy każde wystąpienie konceptu A jest wystąpieniem konceptu B ($A \subseteq B$),
- równości konceptów – czy koncept A równa się konceptowi B ($A = B$) oraz
- rozłączności konceptów – czy koncepty A i B są rozłączne ($A \cap B = \varnothing$).

Drugi zakres obejmuje:

- sprawdzenie wystąpienia – czy dane indywiduum jest wystąpieniem specyficznego konceptu,
- sprawdzenie zgodności indywiduów z hierarchią konceptów – czy indywiduum jest wystąpieniem pewnego konceptu w hierarchii konceptów,
- pobieranie indywiduów danego konceptu – znalezienie wszystkich indywiduów będących wystąpieniami danego konceptu oraz
- klasyfikację indywiduów – znalezienie najbardziej specyficznego konceptu, którego wystąpieniem jest zadane indywiduum.

Jednak logika opisowa ma również swoje wady. Przede wszystkim charakteryzuje się ograniczoną mocą budowania zapytań zawężoną do zdefiniowanych możliwości wnioskowania i ograniczoną zdolnością tworzenia nowych konceptów. Również możliwości obliczeniowe przetwarzania zapytań są nieporównywalnie mniejsze niż w relacyjnych lub obiektowo – relacyjnych bazach danych. Nawet rozszerzenie logiki opisowej o reguły Horna tylko w niewielkim stopniu te wady przezwycięża [2].

Analiza istniejących rozwiązań oraz dostępnych na rynku systemów pozwala zaproponować kierunek badań nad problemem integracji danych łączący w sobie moc wyrażania i przetwarzania zapytań relacyjnych i obiektowo – relacyjnych baz danych z mocą modelowania i wnioskowania semantycznej hierarchii wiedzy dziedzinowej dostarczanej przez logikę opisową. Rozwiązanie takie wydaje się dawać największe perspektywy stworzenia praktycznego systemu integracji danych dostosowanego do współpracy z dużą liczbą źródeł wiedzy, bogatym modelem dziedzinowym i dużą liczbą danych.

BIBLIOGRAFIA

[1] Levy A.Y.: *Logic based techniques in data integration*, Kluwer Publishers 2000.

[2] Baader F.A., McGuness D.L., Nardi D., Patel-Schneider P.F.: *The Description Logic Handbook: Theory, implementation, and applications*, Cambridge University Press, 2003.

[3] Catarci T., Lenzerini M.: *Representing and Using Interschema Knowledge in Cooperative Information System*, International Journal of Intelligent & Cooperative Information Systems, 1993.

[4] Levy A.Y., Srivastava D., Kirk T.: *Data Model and Query Evaluation in Global Information Systems*, Journal of Inteligent Information Systems, Special Issue on Networked Information Discovery and Retrieval, 5(2), 1991.

[5] Arens Y., Knoblock C.A., Shen W.: *Query Reformulation for Dynamic Information Integration*, Journal of Intelligent Information Systems, 1996.
[6] Arens Y., Chee C.Y., Hsu C., Knoblock C.A.: *Retrieving and integrating data from multiple information sources*, International Journal of Intelligent & Cooperative Information Systems, 1993.
[7] Lattes V., Rousset M.-C.: *The use of CARIN language and algorithms for Information Integration: the PICSEL project*, W: Proceedings of the ECAI-98 Workshop on Intelligent Information Integration, 1998.
[8] Donini F. M., Lenzerini M., Nardi D., Schaerf A.: *A Hybryd System with Datalog and Concept Languages*, Trends In Artificial Intelligence, numer LNAI 549, 1991.
[9] Levy A.Y., Rouset M.-C.: CARIN: *A Representation Language Combining Horn rules and Description Logics*, W: Proceedings of the European Conference on Artificial Intelligence, Budapest, Hungary, 1996.
[10] Levy A.Y.: *The Information Manifold Approach to Data Integration*, IEEE Intelligent Systems, numer 13, 1998.

APPLICATIONS OF RELATIONAL DATABASES AND DESCRIPTION LOGICS IN DATA INTEGRATION SYSTEMS

Summary

Data integration of multiple heterogeneous information sources is one of the fundamental problems of knowledge management. The state-of-art of knowledge management systems shows that the most promising practical solution of this problem is to use relational model for semantic notation of knowledge bases. The article presents the most popular approaches to the solution of the data integration problem: standard approach, standard approach extended with domain knowledge and hybrid approach. There are described both advantages and disadvantages of individual solutions as well as data integration systems that exist on the market with the analysis of their usefulness in the real business situations. The aim of the article is to analyze abilities of these solutions focusing on the usage for semantic notation of knowledge bases in relational model.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Jacek Lebiedź

Katedra Inżynierii Wiedzy, Politechnika Gdańska

OKREŚLANIE PODOBIEŃSTWA KSZTAŁTU OBIEKTU DO LITERY LUB CYFRY [1]

Streszczenie

Referat prezentuje różne metody określania na ile wydzielony z obrazu obszar składa się z linii dowolnej grubości. Wyniki tych metod można interpretować jako stopień podobieństwa badanych figur do takich kształtów jak litery czy cyfry bez odnoszenia się do konkretnego alfabetu. Najlepsze z metod znajdą zastosowanie przy lokalizacji napisów dla ręcznego aparatu do odczytywania napisów przeznaczonego dla osób niewidomych (projekt RADONN) oraz przy pomiarze jakości polepszenia obrazu dla potrzeb automatycznego rozpoznawania maszynopisów (projekt MEMORIAL).

1. WSTĘP

Znaki alfanumeryczne czyli litery i cyfry charakteryzują się specyficznym kształtem. Człowiek bez problemu rozróżnia tego typu znaki od innych kształtów nawet jeśli pochodzą one z nieznanego mu alfabetu (chociażby pismo gruzińskie – mchedruli). Można zatem przyjąć, że pewną grupę kształtów daje się określić jako „literopodobne". Innymi słowy istnieją pewne cechy kształtów określające podobieństwo do liter. W artykule tym spróbujemy ustalić te cechy.

Umiejętność rozpoznawania kształtów „literopodobnych" wydaje się być bardzo przydatna. Może ona pozwolić na przykład na wyławianie ze zdjęć obszarów „podejrzanych" o bycie literami i przekazanie ich systemowi OCR (ang. *optical character recognition*) do interpretacji. Znane są co prawda metody wyodrębniania tekstów z obrazów polegające na poszukiwaniu regularności w położeniu i wielkości sąsiadujących ze sobą kształtów, pozwalają one jednak na wykrywanie jedynie napisów składających się z przynajmniej trzech liter [2]. Jedno- lub dwucyfrowy numer tramwaju czy budynku jest więc poza zasięgiem tego typu metod. Tam zatem gdzie jest potrzeba wyłapywania nawet pojedynczych liter konieczny jest algorytm oceny podobieństwa kształtu do litery. Z taką sytuacją mamy do czynienia w przypadku wspomagającego niewidomego ręcznego aparatu do odczytywania napisów RADONN, którego pomysł zarysowano w [1].

[1] sfinansowano w części z funduszy 5.PR UE MEMORIAL / IST-2001-33441

Ocena podobieństwa kształtu do litery bądź cyfry przydatna jest również przy ocenie jakości procesu automatycznego rozpoznawania maszynopisów. Na proces taki składa się kilka etapów, wśród których wyróżnić należy uszlachetnianie obrazu celem podwyższenia skuteczności rozpoznania przez system OCR [3]. Uszlachetnienie to można by oceniać porównując podobieństwo kształtów znaków do liter przed i po uszlachetnieniu. Potrzeba tak zrodziła się podczas realizacji w ramach projektu MEMORIAL zestawu narzędzi do tworzenia dokumentów cyfrowych z zapisów osobowych [8].

2. KLASYCZNE MIARY KSZTAŁTU

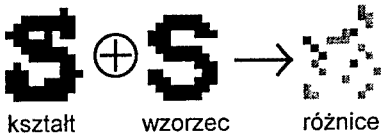
W literaturze znane są rozmaite miary służące do opisu kształtu [4, 6, 7, 9]. Pozwalają one na rozróżnianie rozlicznych figur geometrycznych. Warto zatem sprawdzić, czy któreś z nich nie wystarczą do oceny podobieństwa zadanego kształtu do litery bądź cyfry.

2.1. Podobieństwo map bitowych

Najprostszym sposobem oceny kształtu jest porównanie położenia jego pikseli z mapami bitowymi grupy zadanych wzorców (rys. 2.1). Minimum z liczb różniących się pikseli określa miarę podobieństwa (a właściwie niepodobieństwa, gdyż wartość zerowa oznacza idealną zgodność). Miarę tę opisuje wzór:

$$\varepsilon = \min_k\{\sum_{x,y}|f(x,y)-f_k(x,y)|\}, \tag{2.1}$$

gdzie f i f_k oznaczają funkcje charakterystyczne odpowiednio badanej figury i k-tego wzorca.



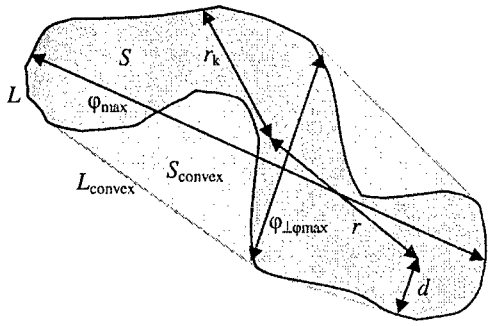
Rys. 2.1. Porównanie mapy bitowej analizowanej figury ze wzorcem litery 's'.

Wadą tego podejścia jest potrzeba wcześniejszego przygotowania odpowiednich wzorców, co w naszym przypadku sprowadza się do konieczności określenia wzorcowego kształtu poszczególnych liter wielkich i małych oraz cyfr (ponad 60 obiektów dla jednego kroju znaków). Ponieważ przesunięcie figury względem badanego wzorca może fałszować wyniki, więc istotne jest tu określenie odpowiedniości porównywanych map bitowych. Problem ten można obejść porównując kształt nie tylko ze wzorcami, ale również z ich translacjami w różnych kierunkach. Wydłuża to niestety kilkukrotnie czas obliczeń. Podejście to jest ponadto bardzo czułe na zmianę rozmiaru i obrót, wymagając w praktyce wzorców dla różnych rozdzielczości i orientacji.

Porównywanie map bitowych należy uznać za nieprzydatne do odczytywania napisów (różne położenie, rozmiary i kroje znaków). W przypadku jednak oceny jakości znaków zeskanowanych maszynopisów (ustalone położenie, rozmiar i krój znaków) podejście to z powodzeniem mogłoby być zastosowane.

2.2. Współczynniki kształtu

Współczynniki kształtu opisują różne cechy geometryczne figur. Najprostsze z nich to spoistość (zwartość) $4\cdot \pi\cdot S / L^2$, kolistość $4\cdot S / \pi\cdot \varphi_{max}{}^2$ i centryczność $\varphi_{max} / \varphi_{\perp\varphi max}$, gdzie S oznacza pole figury (liczba jej pikseli), L – długość jej konturu (liczba pikseli figury sąsiadujących poprzez bok z pikselami spoza figury), φ_{max}, $\varphi_{\perp\varphi max}$ – długości maksymalnej średnicy figury i średnicy prostopadłej do niej (rys. 2.2). Innymi znanymi współczynnikami kształtu są wypukłość L_{convex} / L (L_{convex} – długość konturu uwypuklenia figury), wypełnienie S / S_{convex} (S_{convex} – pole uwypuklenia figury), współczynnik Blaira-Blissa $S / (2\cdot \pi\cdot \sum r^2)^{1/2}$ (r – odległość piksela od środka ciężkości figury), współczynnik Danielssona $S^3 / (\sum d)^2$ (d – najmniejsza odległość piksela od konturu figury), współczynnik Haralicka $\sum r_k / (L\cdot \sum r_k{}^2\text{-}1)^{1/2}$ (r_k – odległość piksela konturu od środka ciężkości figury) [4, 6, 7].



Rys. 2.2. Ilustracja poszczególnych wielkości występujących we wzorach współczynników kształtów.

Współczynniki kształtu pozwalają na rozróżnianie figur, nawet liter między sobą [6]. W większości są mało wrażliwe na przesunięcia, obroty i zmiany rozmiaru [7]. Powszechnie stosowane współczynniki (w tym tutaj przytoczone) nie dają jednak satysfakcjonującego kryterium klasyfikacji kształtów na „literopodobne" i pozostałe. Litery i cyfry mają na tyle odmienne kształty, że potrzebne są bardziej wyrafinowane kryteria.

2.3. Momenty geometryczne

Momenty geometryczne można traktować jako specyficzne współczynniki kształtów. Bazują one na dwuwymiarowym centralnym momencie rzędu $p+q$ z funkcji charakterystycznej f opisującej figurę. Moment ten $M_{p,q}$ i odpowiadający mu moment znormalizowany $N_{p,q}$ zadane są wzorami:

$$M_{p,q} = \sum\sum (x\text{-}\mathrm{E}x)^p\cdot (y\text{-}\mathrm{E}y)^q\cdot f(x,y), \tag{2.2}$$

$$N_{p,q} = M_{p,q} / M_{0,0}{}^{(p+q)/2+1}, \tag{2.3}$$

gdzie symbol E oznacza wartość średnią.

Momenty geometryczne takie jak: $M_1 = N_{2,0} + N_{0,2}$, $M_2 = (N_{2,0} - N_{0,2})^2 + 4\cdot N_{1,1}{}^2$, $M_3 = (N_{3,0} - 3\cdot N_{1,2})^2 + (N_{0,3} - 3\cdot N_{2,1})^2$, $M_4 = (N_{3,0} + N_{1,2})^2 + (N_{0,3} + N_{2,1})^2$, $M_7 = N_{2,0}\cdot N_{0,2} - N_{1,1}{}^2$, $M_8 = N_{3,0}\cdot N_{1,2} + N_{2,1}\cdot N_{0,3} - N_{1,2}{}^2 - N_{2,1}{}^2$ są mniej wrażliwe na przesunięcia, obroty i zmiany rozmiaru (zwłaszcza M_1 i M_7) niż tradycyjne współczynniki

kształtu [7]. Mimo to nie pozwalają na satysfakcjonujący podział kształtów na „literopodobne" i pozostałe. Różnorodność liter i cyfr nie daje się prosto ujednolicić momentami geometrycznymi.

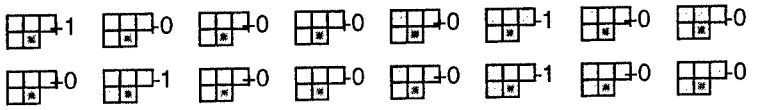
2.4. Miary gładkości konturu

Kontur litery bądź cyfry powinien być stosunkowo gładki. Obrys drzewa lub chmury (na zdjęciu) czy też kleksa lub plamy (na maszynopisie) jest zwykle postrzępiony. Umiejętność oceny gładkości konturu mogłaby niewątpliwie ułatwić odróżnianie liter od innych obiektów. Do oceny gładkości konturu można wykorzystać wymiar fraktalny konturu (sprawdzając, czy jest bliski jedności), analizę harmoniczną konturu (ustalając, czy występuje przewaga małych częstotliwości) lub badanie kodu łańcuchowego konturu (szacując na ile małe są różnice między sąsiednimi elementami) [6].

Metody te jednak są dość złożone i – co więcej – nie mogą stanowić samodzielnego kryterium oceny podobieństwa do litery, gdyż kwadrat czy długa linia mają gładki kontur, a przecież nie są literami. Tym kryterium można tylko odrzucić kształty „postrzępione".

2.5. Charakterystyki topologiczne

Najbardziej chyba znaną charakterystyka topologiczną jest liczba Eulera. Równa jest ona liczbie rozłącznych fragmentów kształtu pomniejszonej o liczbę dziur. Dla figury dyskretnej (zbudowanej z pikseli) można ją wyrazić jak liczba narożników pikseli (jeden narożnik może należeć do czterech pikseli) minus liczba boków pikseli (jeden bok może należeć do dwóch pikseli) plus liczba pikseli [5]. Dzięki temu liczbę tę można bardzo prosto wyliczyć w jednym przejrzeniu obrazu, modyfikując dla każdego piksela liczbę Eulera wyznaczoną dla odczytanego już fragmentu figury o wartość wynikającą z dołączenia aktualnie analizowanego piksela do zbioru odczytanych już pikseli. Algorytm ten daje się też opisać w kategorii wzorców sąsiedztw (rys. 2.3).



Rys. 2.3. Wzorce odczytanych wcześniej (z góry i z lewa) sąsiedztw piksela figury, a sposób modyfikacji liczby Eulera (początkowa wartość: 0).

W przypadku obszarów spójnych liczba Eulera wynosi 1 minus liczba dziur. Liczba dziur jest zatem równa 1 minus liczba Eulera. Tym prostym algorytmem można zatem określić liczbę otworów wewnątrz figury. Dla cyfr arabskich i liter łacińskich liczba otworów może wynosić jedynie 0 (np. dla '2', 'W'), 1 (np. dla '6', 'P') lub 2 (dla '8' i 'B'). Oczywiście liczba otworów czy liczba Eulera może stanowić tylko kryterium pomocnicze przy ustalaniu podobieństwa do liter i cyfr (np. dla '♠' są takie same jak dla '2' czy 'W').

2.6. Grafowe własności szkieletu

Przy ocenie podobieństwa do liter i cyfr pomocne mogą być analizy szkieletu figury. Szkieletem nazywamy wynik ścienienia figury do grubości jednego piksela. Uzyskać to można poprzez cykliczne odrzucanie (erozja) takich pikseli konturu (czyli sąsiadujących bokiem z pikselami spoza zbioru), które znajdują się w konturze tylko z jednej strony.

Na szkielet figury składają się gałęzie i rozgałęzienia. W tym sensie szkielet przypomina graf. Można zatem analizować szkielet jako graf licząc liczbę jego wierzchołków (węzłów), liczbę krańców (węzłów z 1 gałęzią), liczbę rozgałęzień (węzłów z $n > 1$ gałęziami), liczbę krawędzi (gałęzi) czy w końcu liczbę pętli równą zgodnie z równaniem Eulera liczbie krawędzi pomniejszonej o liczbę węzłów i powiększonej o 1. W analizie grafowej warto rozważać długość poszczególnych krawędzi, gdyż krótsze z nich mogą być tylko zaburzeniami kształtu.

Grafowa analiza szkieletu jest dość skomplikowana. Samo znalezienie szkieletów figur wymaga wielokrotnego przeglądania obrazu. Liczenie węzłów i gałęzi jest także złożone.

2.7. Statystyka odległości pikseli od brzegu

Podstawą tego podejścia jest odwzorowanie odległości euklidesowej EDM (ang. *Euclidean Distance Map*), w którym każdemu pikselowi figury przypisana jest najmniejsza odległość od pikseli spoza zbioru. Odwzorowanie to daje się stosunkowo prosto liczyć w dwóch przebiegach w przeciwnych kierunkach [6]. Po jego wyznaczeniu można wartości przypisane pikselom poddać statystycznej obróbce. I tak na przykład w przypadku liter i cyfr histogramy wartości EDM powinny być stosunkowo regularne (rys. 2.4).



Rys. 2.4. Histogramy wartości EDM dla liter 'S' i 'O'.

Analiza charakterystyk statycznych odległości piksela od brzegu wydaje się dość obiecująca. Lepsze rezultaty powinna dać jednak analiza innego nieco odwzorowania zaproponowanego w ostatnim podpunkcie 2.9 tego rozdziału. Z tego też powodu zagadnienia tego nie będziemy tu rozwijać.

2.8. Statystyka szerokości pisaka

Litery i cyfry powstały jako znaki pisane ręcznie. W związku z tym ich kształt ma postać linii pewnej szerokości wynikającej z grubości przyrządu piszącego. Stąd również kształt liter i cyfr generowanych automatycznie (drukowanych) ma zwykle charakter linii stałej szerokości (przynajmniej w pewnym przybliżeniu). Oszacowanie tej szerokości (ang. *fiber width*) i jej rozrzutu (wariancji) na całej drodze hipotetycznego pisaka oraz wyznaczenie długości tej drogi (ang. *fiber length*) pozwala na sformułowanie kryterium oceny podobieństwa kształtu do litery lub cyfry.

Średnia grubość pisaka w_f może być wyliczana na podstawie odwzorowania EDM wykonanego dla szkieletu. Podejście to wymaga znalezienia szkieletu, więc jest dość skomplikowane. Inną ewentualnością jest obliczenie liczby erozji powodujących zniknięcie figury. Złożoność tej metody jest jednak równie duża.

Jako długość drogi hipotetycznego pisaka l_f można przyjąć długość szkieletu. W najprostszym przypadku długość tę wyrazić można liczbą pikseli szkieletu. Taka wartość może jednak dość mocno odbiegać od długości idealnej linii pokrywającej szkielet. Dlatego też lepiej stosować wówczas albo wzór:

$$l_f = 0{,}948 \cdot l_b + 1{,}34 \cdot l_n , \tag{2.4}$$

gdzie l_b i l_n oznaczają liczbę sąsiedztw w konturze odpowiednio poprzez bok i poprzez narożnik [6], albo bazując na oszacowaniach na pole figury (S) i długość jej konturu (L):

$$S = w_f \cdot l_f, \quad L = 2 \cdot (w_f + l_f), \tag{2.5}$$

wzory [6]:

$$l_f = \tfrac{1}{4} \cdot (L - (L^2 - 16 \cdot S)^{1/2}), \quad w_f = S / l_f. \tag{2.6}$$

Od podejścia takiego można oczekiwać stosunkowo wysokiej skuteczności. Jest ono jednak dość złożone obliczeniowo, dlatego też warto zarzucić je na korzyść bardziej efektywnej i do tego zbliżonej koncepcyjnie metody opisanej w następnym podpunkcie.

2.9. Statystyka pokrycia maksymalnymi kwadratami

Metoda opisana w tym podpunkcie łączy w sobie podejścia opisane w poprzednich dwóch podpunktach. Bazuje ona na zbliżonym nieco do EDM odwzorowaniu, które nazwać by można odwzorowaniem maksymalnego kwadratu MSM (ang. *maximal square map*). Każdemu pikselowi figury w tym odwzorowaniu przypisany jest bok największego kwadratu zawierającego ten piksel i w całości leżącego w figurze. Odwzorowanie to wyznacza więc rozmiar najgrubszego pisaka, który mógł narysować dany piksel. Teoretycznie zamiast boków największych kwadratów lepiej by było rozważać średnice największych kół, jednak ze względu na złożoność obliczeń i przewidywanie niewielkich różnic wyników podejścia z kołami zaniechano.

Okazuje się, że MSM wylicza się podobnie jak EDM. Potrzebne są tu na razie trzy przebiegi, choć autor podejrzewa istnienie efektywniejszego algorytmu. Niezależnie od wyników poszukiwań wydajniejszego algorytmu, metodę tę należy uznać za zdecydowanie szybszą od opisanej w poprzednim punkcie.



Rys. 2.5. Zdjęcie wykorzystane do testów: a) oryginał (fot. Piotr Zabłocki), b) wynik segmentacji.

Dla kształtu mającego charakter linii stałej szerokości odwzorowanie MSM powinno znakomitej większości pikseli przypisać zbliżone (jeśli nie takie same) wartości. Histogram wartości MSM dla takich kształtów powinien mieć zatem wyraźne maksimum oraz niewielki rozrzut. Średnie odchylenie (dyspersja bądź wariancja) wyników MSM może więc określać podobieństwo figury do litery lub cyfry rozumianej jako kształt o charakterze linii. Długość takiej linii ma zwykle ograniczony charakter w przypadku liter i cyfr. Obliczyć ją można jako iloraz liczby pikseli w figurze (pole figury) przez kwadrat średniej wartości MSM (pole średniego kwadratu). Przynależność tak wyznaczonej długości do

przedziału potencjalnych długości znaków alfanumerycznych może stanowić kryterium uzupełniające (w stosunku do średniego odchylenia) podobieństwa kształtu do liter i cyfr.

Wykonane przez autora testy zaproponowanej metody potwierdziły przydatność tego podejścia do oceny podobieństwa badanych kształtów do liter i cyfr. Testów dokonano na różnych obrazach: zarówno na zdjęciach zrobionych z perspektywy przechodnia na ulicy (rys. 2.5, 2.6), jak i na „skanach" archiwalnych maszynopisów (rys. 2.7, 2.8).



Rys. 2.6. Ocena podobieństwa poszczególnych segmentów z rysunku 2.5 b) do liter lub cyfr w oparciu o kryteria bazujące na MSM: a) jasność proporcjonalna do ilorazu dyspersji MSM przez średnią wartość MSM (najciemniejsze są litery), b) jasność zależna od długości hipotetycznej linii składającej się na figurę, tzn. jasność proporcjonalna do $h \cdot (2\text{-}h)$, gdzie h jest siedemnastokrotnie pomniejszonym ilorazem liczby pikseli przez kwadrat średniej wartości MSM (najciemniejsze są litery; niektóre litery znikły, gdyż stanowią z sąsiadami pojedynczy segment o zbyt dużej długości).

Konzentrationslager Stutthof
Kommandantur
Konzentrationslager Stutthof
Kommandantur
POLITISCHE ABTEILUNG
POLITISCHE ABTEILUNG
Betr.: Eiweisung von Häftlingen
Bezug: O H N E
An das
Schutzhaftlager

Rys. 2.7. Maszynopis wykorzystany do testów (po binaryzacji).

a)
Konzentrationslager Stutthof
Kommandantur
Konzentrationslager Stutthof
Kommandantur
POLITISCHE ABTEILUNG
POLITISCHE ABTEILUNG
Betr.: Eiweisung von Häftlingen
Bezug: O H N E
An das
Schutzhaftlager

b)
onzentrationsl er Stutthof
Kommandantur
Konzentrationsl er Stutthof
Kommandantur
POLITISCHE ABTEILUNG
POLITISCH ABTEILUNG
Betr.: Eiweisung von Häftlingen
Bezug O H N E
An das
Schutzhaftlager

Rys.2.8. Ocena podobieństwa poszczególnych segmentów z rysunku 2.7 do liter lub cyfr w oparciu o kryteria bazujące na MSM: a) i b) jak w podpisie pod rys. 2.6 (warto zwrócić uwagę na odrzucenie długich podkreśleń przez kryterium z rys. b).

3. ZAKOŃCZENIE

Artykuł omawia różne podejścia pozwalające ocenić kształt obiektu. Dokonuje przede wszystkim analizy przydatności poszczególnych technik do oceny podobieństwa kształtu do litery lub cyfry. Ponieważ rezultaty tej analizy nie rokowały nadziei na efektywne wykorzystanie znanych metod do takiej oceny, zaproponowano oryginalną metodę bazującą na zdefiniowanym w artykule odwzorowaniu maksymalnego kwadratu MSM. Wyniki testów nowej metody potwierdziły jej wysoką przydatność do określania podobieństwa kształtu do litery lub cyfry.

BIBLIOGRAFIA

[1] Kowalik R.: *RADONN – ręczny aparat do odczytywania napisów*. Opracowanie wewnętrzne WETI Politechnika Gdańska, Gdańsk 2002.
[2] Krzywieński M.: *Wykrywacz tekstu dla osób niewidomych – praca dyplomowa*. Politechnika Gdańska, Gdańsk 2003
[3] Lebiedź J., Podgórski A., Szwoch M.: *Pomiar jakości procesu automatycznego rozpoznawania maszynopisów*. W: Technologie informacyjne (materiały konferencyjne), Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej nr 2, Gdańsk 2003, str. 629÷638.
[4] Malina W., Ablameyko S., Pawlak W.: *Podstawy cyfrowego przetwarzania obrazów*. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002.
[5] Román-Roldán R., Gómez-Lopera J. F., Atae-Allah Ch., Martínez_Aroza J., Luque-Escamilla P. L.: *A measure of quality for evaluating methods of segmentation and edge detection*. W: Pattern Recognition 34, 2001, str. 969÷980.
[6] Russ J. C.: *The Image Processing Handbook (Fourth Edition)*. CRC Press, Boca Raton 2002.
[7] Tadeusiewicz R.: *Systemy wizyjne robotów przemysłowych*. WNT, Warszawa 1992.
[8] Wiszniewski B.: *Projekt IST-2001-33441-MEMORIAL: zestaw narzędziowy do tworzenia dokumentów cyfrowych z zapisów osobowych*. W: Technologie informacyjne (materiały konferencyjne), Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej nr 2, Gdańsk 2003, str. 649÷656.
[9] Zhang D., Lu G.: *Review of shape representation and description techniques*. W: Pattern Recognition 37, 2004, str. 1÷19.

SIMILARITY EVALUATION OF OBJECT SHAPE TO LETTER OR DIGIT

Summary

The paper presents different approaches to evaluation of shape similarity to letter or digit. The new method basing on maximal square map (MPM) has been proposed. The results of this method are shown and discussed.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Jacek Lebiedź, Marcin Stankiewicz

Katedra Inżynierii Wiedzy, Politechnika Gdańska

POMIAR WYDAJNOŚCI SYSTEMÓW GRAFIKI TRÓJWYMIAROWEJ [1]

Streszczenie

Tworząc interakcyjny system graficzny należy mieć na uwadze wydajność zarówno sprzętu, jak i obsługującego go oprogramowania. Zaniedbanie analizy możliwości hardware'u i software'u grozi brakiem wydolności systemu i w efekcie koniecznością kosztownej modyfikacji sprzętu lub pracochłonnej (ze względu na przebudowę kodu) wymiany biblioteki graficznej. Oszacowanie wydajności systemu graficznego wydaje się zatem niezbędne w początkowym etapie jego wytwarzania. W oszacowaniu tym należy jednak uwzględniać specyfikę operacji graficznych tworzonego systemu. Z tego też powodu popularne benchmarki nie zawsze pozwalają na trafną ocenę wydajności. Referat przedstawia własną propozycję wielokryterialnego pomiaru wydajności systemu graficznego.

1. WSTĘP

Potrzeba mierzenia wydajności systemu graficznego pojawiła się stosunkowo niedawno. Wymusił ją w ostatnich latach szybki rozwój aplikacji różnego typu (od pakietów biurowych, poprzez gry komputerowe, aż po specjalistyczne narzędzia graficzne), zużywających coraz to więcej zasobów i mających określone wymagania pod względem jakości i szybkości wyświetlanej grafiki [3].

Na rynku można znaleźć wiele różnorakich akceleratorów graficznych. Mnogość ta jest przejawem konkurencji producentów kart graficznych. Oferują oni ogromną liczbę różnych produktów, charakteryzujących się wieloma parametrami. Dodają obsługę różnorodnych udogodnień, ulepszają jakość wyświetlanej grafiki, zwiększają szybkość generacji obrazu. Wydajniejsza karta oczywiście zwykle więcej kosztuje. Nie wszystkie opcje są jednak każdemu potrzebne. Kupując kartę graficzną powinniśmy zatem wybrać produkt najbardziej odpowiadający zadaniom, do jakich będzie używany wyposażony w nią komputer. Warto wówczas zapoznać się z testami dostępnego na rynku sprzętu.

Przeglądając opracowania dotyczące akceleratorów graficznych możemy natknąć się na wyniki testów ich szybkości mierzone za pomocą tzw. benchmarków czyli programów służących do pomiaru wydajności i stanowiących swoisty wzorzec wykorzystania możliwości testowanego urządzenia. Nierzadko jako benchmarków graficznych używa się specjalnie do tego stworzonych programów. Najpopularniejszymi wśród nich są programy

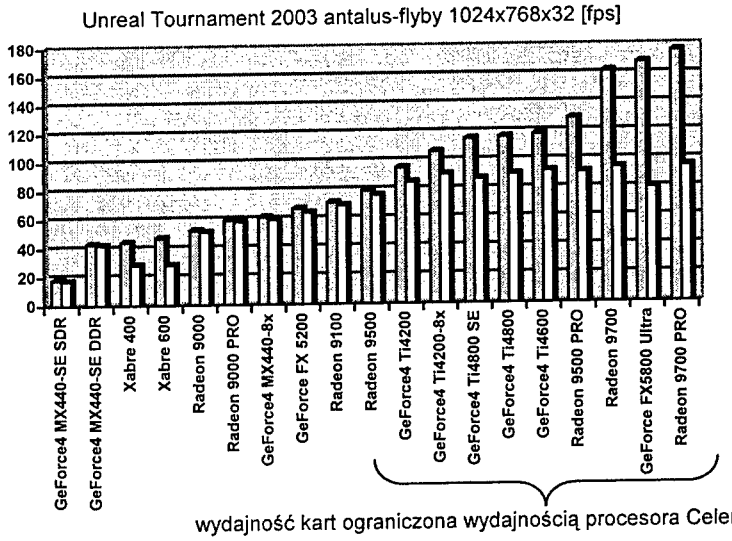
[1] sfinansowano w części ze środków grantu badawczego KBN 4-T11C-004-22

rodziny *3DMark* wyprodukowane przez firmę *FutureMark* (dawniej *MadOnion*). Nie mniej częste jest korzystanie do oceny wydajności z programów, których jedynie efektem ubocznym jest pomiar prędkości i jakości generacji grafiki. Wśród nich niewątpliwie prym wiodą gry komputerowe, jako programy najpełniej wykorzystujące możliwości drzemiące w kartach graficznych.

Programy testujące wydajność komputera możemy podzielić na analityczne i syntetyczne. Pierwsze korzystają z autentycznych fragmentów aplikacji – na przykład gier, czy też edytorów tekstu lub arkuszy kalkulacyjnych. Wynik dokonanego testu informuje o wydajności całego komputera w realnym, typowym zastosowaniu. Natomiast narzędzia przygotowane do przeprowadzania testów syntetycznych zlecają komputerowi wykonanie określonej czynności – na przykład narysowanie 50 milionów trójkątów. Testy syntetyczne są profilowane – pozwalają określić, jak maszyna poradzi sobie z renderowaniem trójwymiarowej grafiki, obliczeniami matematycznymi i innymi zadaniami.

Programy testujące można ponadto podzielić na wydajnościowe (mierzące szybkość przeprowadzania pewnych operacji) oraz sprawdzające. Te ostatnie – określane też mianem jakościowych – pomagają stwierdzić obecność określonych bibliotek systemowych, ocenić jakość obrazu monitora itp.

Analizując wyniki testów warto też pamiętać, że stanowią one ocenę całej konfiguracji sprzętowej komputera. Żaden benchmark nie może przetestować karty graficznej w oderwaniu od środowiska, w którym ona występuje. Jeżeli na przykład szybkość grafiki zależy od użytej rozdzielczości ekranu lub głębi kolorów, to przyczyna tego tkwi nie w akceleratorze graficznym, gdyż do akceleratora przesyłane są te same współrzędne wierzchołków, lecz raczej w prędkości pamięci bądź szyny danych, które muszą radzić sobie z większą ilością danych i mogą spowalniać działanie aplikacji, stanowiąc wówczas wąskie gardło systemu [5]. Zbyt wolny procesor może również spowalniać działanie karty graficznej, co pokazuje rys. 1.1 [1].



Rys. 1.1. Wydajność różnych kart graficznych pracujących na dwóch platformach: Intel Pentium 4 3,06 HT – ciemniejsze słupki i Intel Celeron 1,7 GHz – jaśniejsze słupki (Windows XP, 512 MB DDR RAM, dysk twardy Maxtor Diamond Plus 9 40 GB, płyta główna z chipsetem i 7205) [1].

2. GRY KOMPUTEROWE JAKO BENCHMARKI

Wykorzystanie gier komputerowych jako benchmarków jest zwłaszcza celowe, gdy głównym przeznaczeniem komputera mają być właśnie gry komputerowe. Nie ma co ukrywać, że większość akceleratorów 3D jest kupowana właśnie w celach rozrywkowych. W takich przypadkach prędkość działania karty graficznej podczas obsługi gry, jest oceną miarodajną. Co więcej, dla użytkownika bardziej przekonujące są wyniki podane w klatkach na sekundę, które akcelerator osiągnął podczas gry w *Quake*'a, niż mało mówiąca ocena szybkości w bliżej niezidentyfikowanych jednostkach, jaką otrzymujemy w profesjonalnych benchamarkach 3D.

Najczęściej używanymi w roli benchmarków są gry z perspektywy pierwszej osoby, określane w skrócie mianem gier FPP (ang. *First Person Perspective*), takie jak seria *Quake* czy *Unreal*. Gry tego typu (szczególnie ich najnowsze wersje) charakteryzują się wysokim obciążeniem akceleratora, wynikającym z presji rynku wymuszającego dążenie do uzyskania jak najbardziej skomplikowanego i pociągającego świata gry.

Prędkość działania gier komputerowych stanowi jednak ocenę całego systemu, nie tylko jego strony graficznej. Prędkość ta bowiem nie zależy wyłącznie od grafiki, gdyż na gry komputerowe składa się również dźwięk, sztuczna inteligencja i obsługa innych urządzeń zewnętrznych. Ocena szybkości gry jest więc nie tylko oceną wydajności grafiki, ale oceną całej platformy sprzętowej.

3. SPECJALIZOWANE BENCHMARKI

Najpopularniejszymi programami stworzonymi specjalnie do testowania parametrów platformy graficznej komputera są programy serii *3DMark*. Bazują one na pewnych scenach wzorcowych, dla których wyznaczana jest szybkość (w klatkach na sekundę) ich generacji przy określonych poziomach szczegółowości. Wynikiem końcowym jest odpowiednio ważona suma uzyskanych wartości wyrażona niestety w niewiele mówiących jednostkach o nazwie *3D mark*. Oprócz wyniku ogólnego programy te podają również wyniki testowania poszczególnych elementów generacji obrazu dla pewnych scen uproszczonych, takie jak prędkość wypełniania teksturą (w milionach tekseli na sekundę) czy prędkość wyświetlania wielokątów (w tysiącach trójkątów na sekundę).

3.1. *3DMark 2000*

Obrazy testowe programu *3DMark 2000* stanowią dwie sceny zapożyczone z gier: symulatora śmigłowca bojowego i klasycznej „strzelanki" FPP. Sceny te testowane są dla trzech różnych poziomów szczegółów: niskiego *L* (kilka tysięcy trójkątów), średniego *M* (kilkanaście tysięcy trójkątów) i wysokiego *H* (kilkadziesiąt tysięcy trójkątów). Wynik końcowy wyznaczany jest na podstawie wzoru:

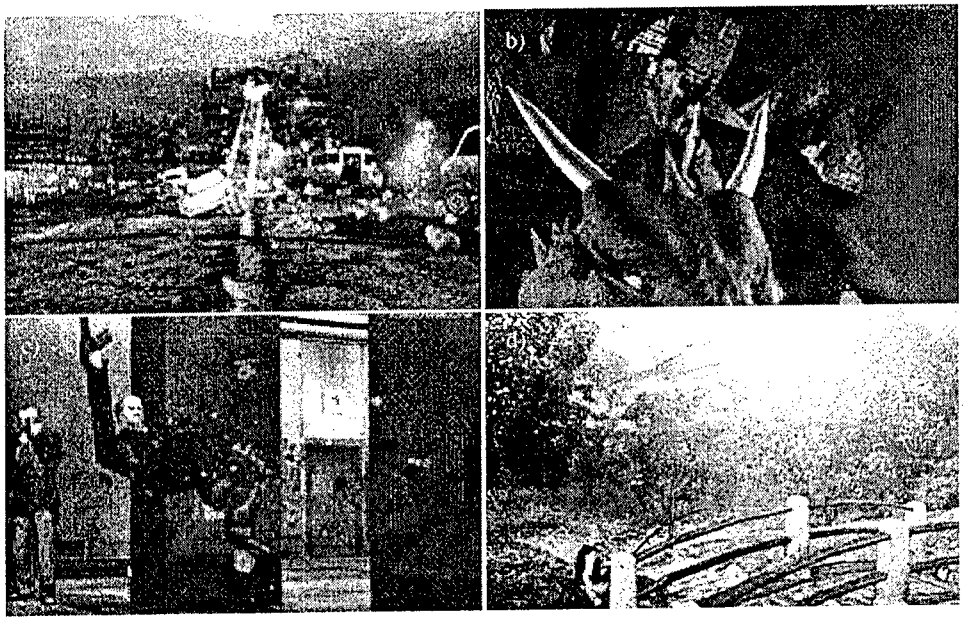
$$R_{3DMark2000} = 12 \cdot (f_{1L}+f_{1M}+f_{1H}+f_{2L}+f_{2M}+f_{2H}), \tag{3.1}$$

gdzie f_{iJ} oznacza prędkość generacji i-tej sceny (symulator śmigłowca, „strzelanka") przy J-tym poziomie szczegółów (niski, średni, wysoki) wyrażoną w klatkach na sekundę [4].

3.2. *3DMark 2001*

Program *3DMark 2001* przeznaczony jest do oceny wydajności kart graficznych sprzętowo zgodnych z *DirectX* 7 lub *DirectX* 8 i wymaga biblioteki graficznej *DirectX* 8.0 udostępniającej ich zaawansowane mechanizmy graficzne, takie jak cieniowanie wierzchołków (ang. *vertex shader*), cieniowanie pikseli (ang. *pixel shader*), duszki punktowe (ang. *point sprites*), wygładzenie pełnoekranowe (ang. *full scene anti-aliasing FSAA*) czy kompresja tekstur. Wykorzystuje on ponadto motor (ang. *engine*) graficzny o nazwie *MAX-FX* firmy *Remedy Entertainment*.

Testy są tutaj przeprowadzane również na scenach z trójwymiarowych gier komputerowych. Sceny te, poza ostatnią, testowane są przy niskim (*L*) i wysokim (*H*) poziomie szczegółów, ostatnia tylko przy wysokim. Pierwsza z nich przedstawia grę *Car chase*, w której wrogie roboty atakują Ziemię. Roboty, samochody i latające obiekty mają tu trzy warstwy tekstur. Każdy obiekt posiada ponadto dynamiczne cienie. Wszelki ruch odbywa się z zachowaniem praw fizyki, nawet siły tarcia opon są wyliczane i wpływają na ruch samochodów. W trybie niskiej jakości jest średnio 33753 trójkątów, natomiast w trybie wysokiej jakości 67597 (rys. 3.1 a).



Rys. 3.1. Kadry z programu *3DMark 2001*: a) *Car chase*, b) *Dragothic*, c) *Lobby*, d) *Nature* [6].

Druga ze scen dotyczy gry *Dragothic*, w której smok atakuje wioskę, a żołnierze odpowiadają ognistymi strzałami. Test ten używa nowego systemu globalnego oświetlenia *Limugraph*. Animacje wykorzystują tu głównie cieniowanie wierzchołków i morfing (rys. 3.1 b). Trzecia scena z gry *Lobby*, przywodząca skojarzenia z filmem *Matrix*, przedstawia walkę Neo w holu ogromnego budynku. Postacie i animacje są zbliżone do występujących w grze *Max Paine*. Przy generacji ludzkich postaci wykorzystano cieniowanie wierzchołków i jednostkę *T&L* (rys. 3.1 c).

Czwarta, ostatnia scena *Nature*, najbardziej realistyczna, ukazuje mężczyznę łowiącego ryby nad jeziorem, otoczonego drzewami, motylami itp. Animacja mężczyzny uzyskana jest przy użyciu morphingu zrealizowanego na bazie techniki programowego cieniowania wierzchołków. Oprócz tego test ten wykorzystuje cieniowanie wierzchołków i odwzorowanie sześcienne (ang. *cube mapping*). Test ten wyznaczany jest tylko dla wysokiego poziomu szczegółów (rys. 3.1 d).

Wynik końcowy testu jest w programie *3DMark 2001* wyznaczany nieco inaczej niż poprzednio, a mianowicie na podstawie wzoru:

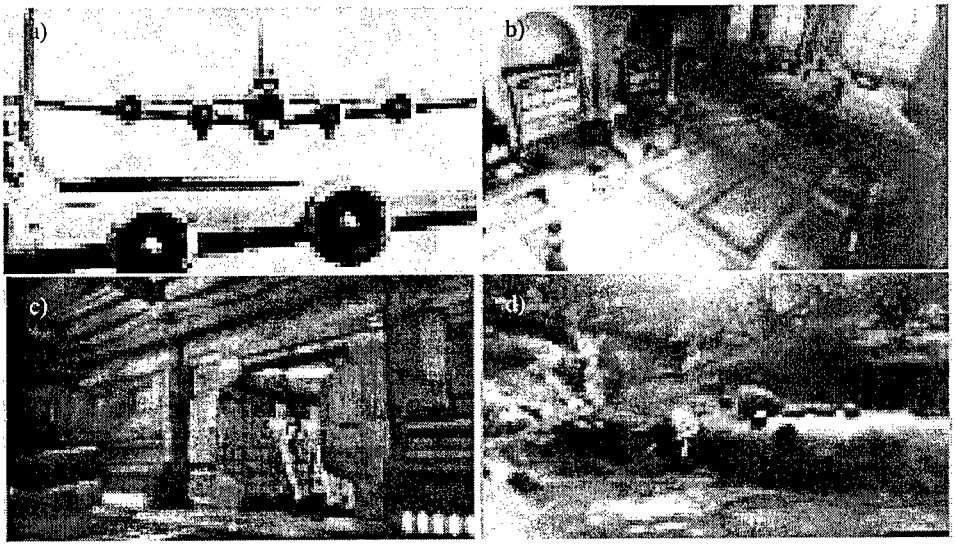
$$R_{3DMark2001} = 10 \cdot (f_{1L} + f_{2L} + f_{3L}) + 20 \cdot (f_{1H} + f_{2H} + f_{3H} + f_{4H}), \tag{3.2}$$

gdzie f_{ij} oznacza prędkość generacji i-tej sceny przy J-tym poziomie szczegółów (fps) [6].

3.3. *3DMark 2003*

Program *3DMark 2003* dedykowany jest z kolei do oceny kart graficznych sprzętowo zgodnych z *DirectX* 9 i dlatego wymaga biblioteki graficznej *DirectX* 9.0 udostępniającej wiele zaawansowanych mechanizmów graficznych. Podobnie jak w poprzedniej wersji testy są przeprowadzane na scenach z czterech trójwymiarowych gier komputerowych. Mimo zastrzeżeń firmy *NVIDIA* jednego z największych producentów kart graficznych co do jakości jego testów [8], benchmark ten wydaje się pewnym następcą swoich dwóch poprzedników.

Pierwsza ze scen przedstawia grę *Wings of Fury* wymagającą jedynie zgodności sprzętowej z *DirectX 7*. Jej tematem jest obrona grupy alianckich bombowców przed atakiem myśliwców wroga. W grze używana jest stosunkowo mała liczba wielokątów, korzysta się jednak z systemów cząsteczkowych (np. smugi dymu, wybuchy). Samoloty są renderowane z czterema warstwami tekstur (rys. 3.2 a).



Rys. 3.2. Kadry z programu *3DMark 2003*:
a) *Wings of Fury*, b) *Battle of Proxycon*, c) *Troll's Lair*, d) *Mother Nature* [2].

Drugi i trzeci test dotyczy gier FPP i RPG (ang. *Role Playing Game*) wymagających zgodności sprzętowej z *DirectX 8*. Akcja gry FPP *Battle of Proxycon* rozgrywa się na statku kosmicznym, którego pokład próbują opanować żołnierze przeciwnika (rys. 3.2 b). W grze RPG *Troll's Lair* poszukiwaczka przygód eksplorując starą pracownię napotyka dwa nieprzyjazne trole, które zmuszają ją do walki (rys. 3.2 c). Podczas generacji scen obu gier szeroko wykorzystuje się cieniowanie wierzchołków i pikseli. Stosowane są tam zaawansowane techniki teksturowania (ang. *normal mapping*) i dynamicznych cieni (ang. *stencil shadowing*). Używane są ponadto efekty głębokości pola (ang. *depth of field*) i smug świetlnych (ang. *bloom*). Sceny *Troll's Lair* bazują na większej liczbie wielokątów, chociażby z powodu zamodelowania poszczególnych włosów głównej bohaterki. Włosy te charakteryzują się wysoką naturalnością ruchu i wyglądu (oświetlenie anizotropowe).

Ostatnia scena *Mother Nature* wymaga sprzętu zgodnego z *DirectX 9* i przedstawia płynący strumień w otoczeniu kołyszących się drzew i trawy. Poszczególne liście drzew i źdźbła traw są modelowane osobno z wykorzystaniem cieniowania wierzchołków. Powierzchnie jeziora i nieba są uzyskane z użyciem cieniowania pikseli.

Ostateczny wynik testu w programie *3DMark 2003* jest wyznaczony jako suma ważona poszczególnych testów wg wzoru:

$$R_{3DMark2003} = 7{,}3 \cdot f_1 + 37 \cdot f_2 + 47{,}1 \cdot f_3 + 38{,}7 \cdot f_4, \tag{3.3}$$

gdzie f_i oznacza prędkość generacji i-tej sceny (fps). Wynik ten przyjmuje wartości od 1000 (CPU 1GHz, karta graficzna *Directx 8*) do 5000 (CPU 5GHz, karta *Directx 9*) i więcej [2].

4. OCENA WYDAJNOŚCI

Wyniki otrzymane w programach oceniających wydajność systemu graficznego teoretycznie mówią nam o tym, jak szybko nasz system jest w stanie wyświetlać grafikę. Nie mając wiedzy na temat algorytmu zastosowanego do oceny wydajności karty graficznej użytkownik nie może stwierdzić na co położono nacisk w ocenie wydajności, a jakie aspekty pominięto. Ocenia więc system porównując jedynie wyniki z tymi, które zostały otrzymane na innych platformach sprzętowych.

Uzyskanie wiarygodnej formuły oceniającej wydajność systemu jest trudnym zadaniem. Ocena musi być obiektywna. Co więcej, każda nowa generacja procesorów graficznych wprowadza nie tylko większą wydajność, rozdzielczość, czy głębię kolorów, ale także dodatkowe technologie renderingu, które poprawiają realizm sceny i zwiększają złożoność obrazu. Najnowsze wersje akceleratorów 3D posiadają na przykład opcje dostępne dotychczas tylko w profesjonalnych rozwiązaniach sprzętowych, takie jak sprzętowa transformacja i oświetlenie (*T&L*), programowalna jednostka cieniowania pikseli czy wygładzanie pełnoekranowe.

Producenci podają wydajność swoich produktów w pikselach bądź tekselach na sekundę. Brak tam jednak właściwych ocen wydajności i ograniczeń akceleratorów, takich jak ilość przesyłanych wielokątów na sekundę, obsługiwane sprzętowo efekty, czy szerokość szyny pamięci. Stąd też wynika potrzeba konstruowania niezależnych benchmarków.

Rzetelne przetestowanie sytemu graficznego wymaga wzięcia pod uwagę takich parametrów jak prędkość, funkcjonalność, jakość wizualizacji i cena. Uwzględniając to spostrzeżenie można sformułować wstępny wzór oceny systemu graficznego:

$$R = w_v \cdot R_v + w_f \cdot R_f + w_q \cdot R_q + w_p \cdot R_p, \tag{3.4}$$

gdzie R_i ($R_i \geq 0$) oznacza ocenę odpowiednio prędkości (v), funkcjonalności (f), jakości wizualizacji (q) i ceny (p), a w_i ($0 \leq w_i \leq 1$, $\sum w_i = 1$) określa wagę z jaką dana ocena wpływa na ocenę całościową R.

Wartości wag w_i powinny być dobierane w zależności od potrzeb użytkownika. Dla graczy najważniejsza jest prędkość, następnie cena, funkcjonalność i jakość renderowanego obrazu. Jakość obrazu i funkcjonalność są z kolei najistotniejszymi cechami dla grafików, w dalszej kolejności zaś prędkość i cena. Programiści natomiast mają jeszcze inne priorytety: funkcjonalność, prędkość, jakość i cena. Należałoby pokusić się o stworzenie specjalnej tabeli podającej wartości wag w_i dla poszczególnych kategorii użytkowników. Wartości te można by ustalić na podstawie uśrednienia wag podanych przez reprezentatywną grupę świadomych przedstawicieli każdej z grup. Tablica 4.1 zawiera jedynie przypuszczenie autorów o postaci takiej tabeli.

Tablica 4.1

Przykładowe wartości wag określające wpływ składowych na całościową ocenę systemu graficznego.

kategoria użytkowników	w_v	w_f	w_q	w_p
gracze	0,4	0,2	0,1	0,3
graficy	0,2	0,3	0,4	0,1
programiści	0,3	0,4	0,2	0,1

W Katedrze Inżynierii Wiedzy na Wydziale Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej trwają obecnie prace nad realizacją metod pomiaru wydajności systemów grafiki trójwymiarowej [6, 7]. Stopień zaawansowania prac pozwala szacować, że zostaną one zakończone jeszcze w tym roku. Prezentowane w dalszej części artykułu wyniki tych prac są na razie dość ogólne, jest jednak zbyt wcześnie na przedstawienie szczegółowych propozycji oceny wydajności.

4.1. Ocena prędkości

W generacji obrazu trójwymiarowego uczestniczą trzy podsystemy: procesora, geometrii i rasteryzacji. Ocena prędkości powinna więc dotyczyć każdego z nich. Pomiary należy przeprowadzać tak, by zmierzyć obciążenie procesora, liczbę wielokątów w scenie i liczbę przetwarzanych pikseli przy stałej liczbie klatek na sekundę (przy wyłączonym oczekiwaniu na pionowy powrót plamki). Testy powinny być realizowane w trybie jednobuforowym.

W wielozadaniowych systemach operacyjnych trudno jest zmierzyć obciążenie procesora. Najprostszym rozwiązaniem jest zmierzenie liczby wolnych cykli procesora (kiedy nie wykonuje żadnej pracy). Im mniej obciążony jest procesor, tym bardziej efektywny jest akcelerator graficzny.

Aby ocenić podsystem geometrii należy wyznaczyć jak szybko są przetwarzane różne konfiguracje prymitywów graficznych (ciągi trójkątów, wachlarze itp.). Można wyróżnić następujące sposoby przetwarzania prymitywów – bezpośredni, tablice, tablice indeksowane, listy wyświetlania. Należy ponadto skontrolować ilość danych przesyłanych wraz z każdym wierzchołkiem. Sprawdzić też warto zachowanie systemu przy różnej mocy oświetlenia oraz w obecności efektów atmosferycznych.

Na ocenę etapu rasteryzacji składa się kilka składników: jakość cieniowania, głębokość bufora głębokości, kanał alfa, multiteksturowanie, tekstury animowane.

4.2. Ocena pozostałych parametrów

Ocena funkcjonalności systemu graficznego powinna uwzględniać liczbę obsługiwanych sprzętowo opcji. Odpowiednie testy powinny być zrealizowane na bazie standardowych funkcji 3D API.

Do oceny jakości wizualizacji należy wykorzystać pewną liczbę standardowych scen obrazowanych w stałej rozdzielczości i przy maksymalnych dostępnych ustawieniach głębokości koloru, precyzji bufora głębokości, wielkości tekstur i innych parametrach mających wpływ na jakość obrazu. Tak otrzymane obrazy powinny stanowić podstawę tej oceny.

Ocena ceny powinna być z kolei opisana jako jakaś malejąca funkcja ceny. Najprostszym przykładem takiej funkcji jest odwrotna proporcjonalność. Wówczas ocenę tę wyrazić by można jako $R_p = \alpha/p$, gdzie p stanowi cenę, a α – współczynnik.

5. ZAKOŃCZENIE

W artykule zaprezentowano różne podejścia do mierzenia wydajności podsystemów grafiki w komputerach. Przy ocenie wyników pomiaru należy oczywiście uwzględnić specyfikę operacji graficznych wykonywanych w systemie. W związku z tym zaproponowano kompleksową metodę oceny wydajności uwzględniającą potrzeby konkretnego użytkownika w postaci specjalnie dobieranych wag. Artykuł omawia też technikę pomiaru poszczególnych parametrów – składników ważonej oceny.

BIBLIOGRAFIA

[1] Budny M.: *Karty graficzne.* W: CHIP 6, 2003, str. 30.
[2] Dhagat M.: *3DMark03: Next Generation 3D Benchmarking (White Paper).* W: *FutureMark Corporation* - http://www.futuremark.com/companyinfo/3dmark03_whitepaper.pdf , 2003.
[3] Foley J. D., van Dam A., Feiner S. K., Hughes J. F., Phillips R. L.: *Wprowadzenie do grafiki komputerowej.* WNT, Warszawa 1995.
[4] Grądzki W., Kolano D.: *System oceny wydajności akceleratorów grafiki – praca dyplomowa.* Politechnika Gdańska, Gdańsk 2001.
[5] Pabst Th.: *3D Benchmarking – Understanding Frame Rate Scores.* W: *Tom's Hardware Guide* – http://www.tomshardware.com/graphic/20000704/, 2000.
[6] Stankiewicz M.: *Ocena wydajności systemu graficznego – praca dyplomowa.* Politechnika Gdańska, Gdańsk 2004 (w przygotowaniu).
[7] Trapp Sz.: *Pomiar wydajności systemów wizualizacji trójwymiarowej – praca dyplomowa.* Politechnika Gdańska, Gdańsk 2004 (w przygotowaniu).
[8] Weinand L.: *3DMark 2003: Talking Back to NVIDIA.* W: *Tom's Hardware Guide* – http://www.tomshardware.com/column/20030219/, 2003.

EFFICIENCY EVALUATION OF 3D GRAPHICS SYSTEMS

Summary

The paper presents proposition of multicriterial evaluation of efficiency for three-dimensional graphics systems. This evaluation uses criteria mechanism for taking account of user needs, different for game players, graphic designers or programmers.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Jacek Lebiedź, Marta Stępniak

Katedra Inżynierii Wiedzy, Politechnika Gdańska

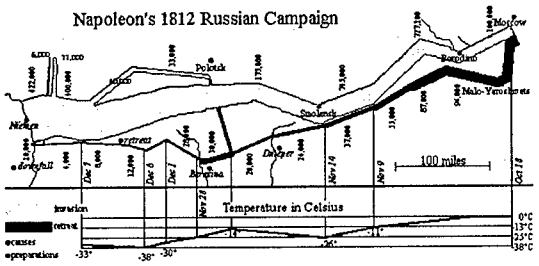
WIELOWYMIAROWA WIZUALIZACJA INFORMACJI

Streszczenie

Metody wizualizacji komputerowej stanowią obecnie podstawowe narzędzie przekazywania człowiekowi informacji, wypełniającej często obszerne zbiory danych, a będącej wynikiem obserwacji, eksperymentów czy też symulacji numerycznych. Zwykle w przypadku danych wielowymiarowych dopiero obejrzenie ich przedstawienia graficznego daje możliwość wyobrażenia sobie jakościowego charakteru zjawiska i uprawnia do stawiania hipotez jego dotyczących. Poznanie specyfiki prezentowanego zjawiska zależy w dużej mierze od wybranej formy graficznej. Referat jest próbą systematyzacji graficznych reprezentacji informacji ze wskazaniem sytuacji, w jakich ich stosowanie daje najlepsze wyniki.

1. WPROWADZENIE

Jednym z cenniejszych elementów, jakie wnoszą komputery nie tylko do badań naukowych, lecz np. również do dydaktyki czy też do analizy wspomagającej podejmowanie decyzji, jest duża łatwość wizualizacji obiektów, procesów i zależności między nimi. Wizualizacja nie jest jednak zagadnieniem nowym, które narodziło się dopiero w dobie komputerów. Znane są w historii przykłady bardzo udanych wizualizacji, chociażby mapa Minarda sprzed bez mała 200 lat uzmysławiająca tragedię wyprawy napoleońskiej na Moskwę w 1812 roku (rys. 1.1) [4, 6, 7, 9]. Wizualizacje te były oczywiście tworzone ręcznie, przez co ich wytwarzanie było stosunkowo żmudne. Dopiero dzięki komputerom stała się możliwa automatyzacja procesu wizualizacji i w konsekwencji duża łatwość jego wykonania.

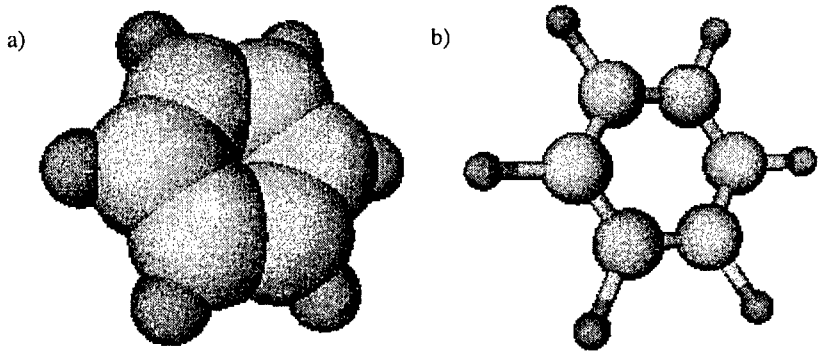


Rys. 1.1. Mapa Minarda przedstawiająca na mapie geograficznej (zaznaczone rzeki i miejsca bitew) kierunek marszu wojsk i ich liczebność (grubość linii) oraz temperaturę powietrza (dolny wykres) [7].

Celem wizualizacji jest kształtowanie obrazu zjawiska w umyśle odbiorcy dla lepszego jego zrozumienia. Wizualizacja powinna zatem prowadzić raczej do odsłaniania tego co ukryte, niż wieść do ukazywania tego co znane. Do uzyskania tego efektu wykorzystuje się głównie wzrok, choć słuch i inne zmysły mogą również odgrywać pewną rolę. Sam termin wizualizacja wywodzi się z łaciny (łac. *visualis* – wzrokowy) i trafił do języka polskiego za pośrednictwem angielskiego czasownika *visualize*, tłumaczonego jako wyobrażanie, uzmysławianie, przypomnienie sobie czegoś, w taki sposób jakbyśmy to aktualnie widzieli lub kształtowanie wizerunku lub wizji czegoś [9]. Wizualizację można zatem najprościej określić jako „przedstawienie zjawiska za pomocą obrazów".

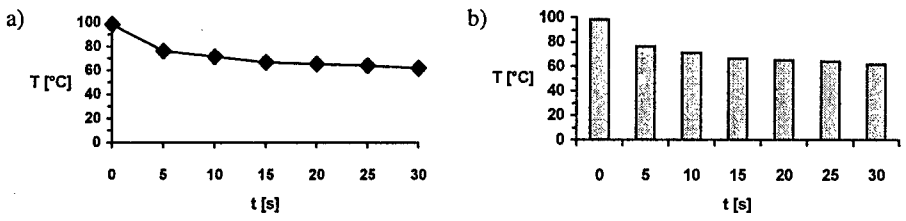
Dzięki takiemu rozumieniu wizualizacji, korzyści z jej stosowania podczas analizy informacji trudne są do przecenienia. Wizualizacja nie tylko upraszcza znajdowanie zależności w informacji, czy też ukazywanie jej znaczących cech, ale wzmaga także proces rozumienia informacji, pozwala na nowe spojrzenie na nią, uwidacznia to co niewidoczne. W konsekwencji wizualizacja jest nieoceniona zwłaszcza w pracy naukowej, gdzie przede wszystkim zwiększa jej wydajność np. wpływając na jakość sterowania symulacją lub pomiarem [6].

Rzeczywiste zjawisko daje się wizualizować na dwa sposoby. Można je zobrazować niejako bezpośrednio, konstruując i ukazując świat wirtualnych obiektów reprezentujących obiekty realne. Obraz taki niejednokrotnie powstaje z użyciem zaawansowanych metod trójwymiarowej grafiki komputerowej. Powszechnie znanym przykładem tego rodzaju wizualizacji jest przedstawienie cząsteczek chemicznych za pomocą kulek (rys. 1.2) [10]. Zjawisko może być również prezentowane pośrednio, przez pryzmat zależności między jego mierzalnymi parametrami. Istotny jest wtedy dobór właściwego rodzaju wykresu do odpowiedniego zobrazowania zależności. Jednym z prostszych przykładów tego typu wizualizacji jest przedstawienie spadku w czasie temperatury wskazywanej przez wyjęty z wrzątku termometr rtęciowy (rys. 1.3) [10].



Rys. 1.2. Przykład wizualizacji naukowej – cząsteczka benzenu H_6C_6:
a) model czaszowy, b) model „kulki i pałeczki" [10].

Przyjmując powyższe kryteria wizualizację możemy podzielić na wizualizację przedstawiającą wirtualny model zjawiska i wizualizację zależności opisujących zjawisko. Pierwszy rodzaj wizualizacji określa się mianem wizualizacji naukowej, drugi zaś wizualizacji danych [6]. Wizualizacja naukowa jest zatem obrazowaniem zjawiska (obiektu lub procesu), wizualizacja danych jest natomiast schematycznym przedstawieniem zależności między danymi.



Rys. 1.3. Przykład wizualizacji danych – zależność temperatury wskazywanej przez wyjęty z wrzątku termometr rtęciowy od czasu jaki upłynął od wyjęcia: a) wykres liniowy, b) wykres słupkowy [10].

2. WIZUALIZACJA NAUKOWA

Wizualizacja naukowa jest graficzną prezentacją rzeczywistego zjawiska. W wielu zastosowaniach spotykamy się ze zjawiskami, których analiza wymaga zaawansowanego i kosztownego sprzętu np. mikroskopów elektronowych, mierników o dużej czułości lub wręcz jest niemożliwa. Znacznie taniej jest użyć do analizy wizualizacji naukowej. Ten rodzaj wizualizacji wykorzystuje się również w analizie zjawisk, w których obiekty są bądź za małe (struktura atomu, molekuły), bądź zbyt odległe (kosmologia, czarne dziury), bądź procesy zachodzą zbyt szybko (uderzenie pocisku w przeszkodę), bądź też zbyt wolno (ewolucja). W związku z tym wizualizacja wykorzystywana jest w wielu dziedzinach nauki, takich jak chemia (cząsteczki chemiczne – rys. 1.2), biologia (synteza kwasu DNA – rys. 2.1), fizyka (przepływ cieczy – rys. 2.2), astronomia (obroty Księżyca wokół Ziemi – rys. 2.4), matematyka (ruch wahadłowy – rys. 2.5), itd. Szczegółowe omówienie wspomnianych tu przykładów wizualizacji naukowej można znaleźć w pracy [10].

Wizualizacja naukowa wykorzystuje różne metody grafiki komputerowej służące do obrazowania dwu- i trójwymiarowego oraz uzyskiwania wrażenia ruchu (animacja), a nawet techniki rzeczywistości wirtualnej. Stosowanie trzech wymiarów (wzbogacone o teksturowanie, cieniowanie i przezroczystość) wzmaga odczucie realizmu, animacja pozwala na przedstawienie zmieniających się parametrów w czasie, a pełną swobodę uzyskujemy w rzeczywistości wirtualnej, gdzie możemy poruszać się między obiektami i umieszczać obserwatora w dowolnym punkcie wirtualnego świata.

Bardzo ważną cechą wizualizacji naukowej jest interaktywność rozumiana jako zdolność sterowania analizowanym zjawiskiem. Możliwość zmieniania wartości parametrów obiektu czy procesu pozwala na wygodne dokonywanie eksperymentów. Możność obserwowania zjawiska z różnych miejsc i w różnym czasie dodatkowo ułatwia analizę prowadzonych doświadczeń. Co więcej wygoda badań potęgowana jest przez atrakcyjne mechanizmy kontrolowania zjawiska i przemieszczania się w obrębie niego zbliżone do interfejsów gier komputerowych.

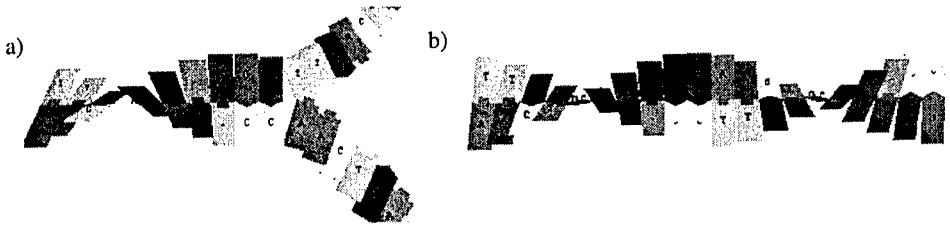
Czas w wizualizacji naukowej pełni istotną rolę. Ze względu na powiązania zjawiska z czasem można wyróżnić cztery rodzaje wizualizacji: wizualizację statyczną (nie uwzględniającą aspektu czasu), statyczną wizualizację czasu (czas prezentowany jako wymiar przestrzeni), dynamiczną wizualizację parametru nie będącego czasem (animacja nie oddająca upływu czasu) oraz wizualizację dynamiczną (animacja ukazująca upływ czasu). Nic nie stoi na przeszkodzie, by podział ten odnieść również do wizualizacji danych, choć dla wizualizacji naukowej wydaje się on bardziej reprezentatywny.

2.1. Wizualizacja statyczna

W wizualizacji statycznej aspekt czasu nie jest uwzględniony. Z taką sytuacją mamy do czynienia najczęściej podczas wizualizacji naukowej obiektów, tam gdzie czas nie jest istotny (np. graf przedstawiający topologię sieci komputerowej; obraz cząsteczki chemicznej - rys. 1.2). Do tego rodzaju wizualizacji zaliczyć należy również znakomitą większość przykładów wizualizacji danych nie związanych z czasem, a więc wszelkiego typu wykresy przedstawiające zależności między parametrami nie będącymi czasem (np. zależność masy elektronu od jego prędkości względem obserwatora; również mapa Minarda może być traktowana jako wizualizacja statyczna, chociaż czas pojawia się na niej w postaci dat pobytu wojsk w poszczególnych miejscach, nie wpływa on jednak na wygląd mapy – rys. 1.1).

2.2. Statyczna wizualizacja czasu

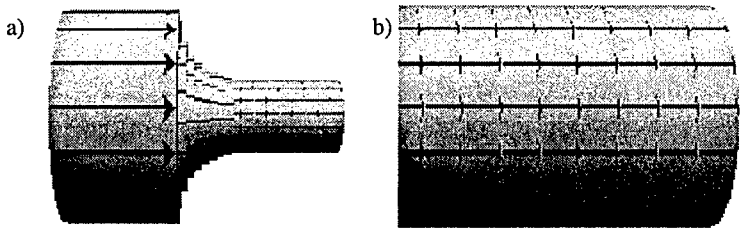
W wizualizacji tego typu czas traktowany jest jako dodatkowy parametr i przedstawiony zostaje w taki sam sposób jak pozostałe atrybuty. Najprostszym chyba podejściem stosowanym w tego typu wizualizacji naukowej jest prezentacja sekwencji obrazów jakiegoś obiektu w różnych chwilach czasu. Przykładem może tu być obrazowanie procesu syntezy kwasu DNA (rys. 2.1). Statyczna wizualizacja czasu ma często również miejsce w wizualizacji danych zależnych od czasu, gdzie przyjmuje postać wykresów z czasem na osi odciętych (np. zależność temperatury od czasu – rys. 1.3).



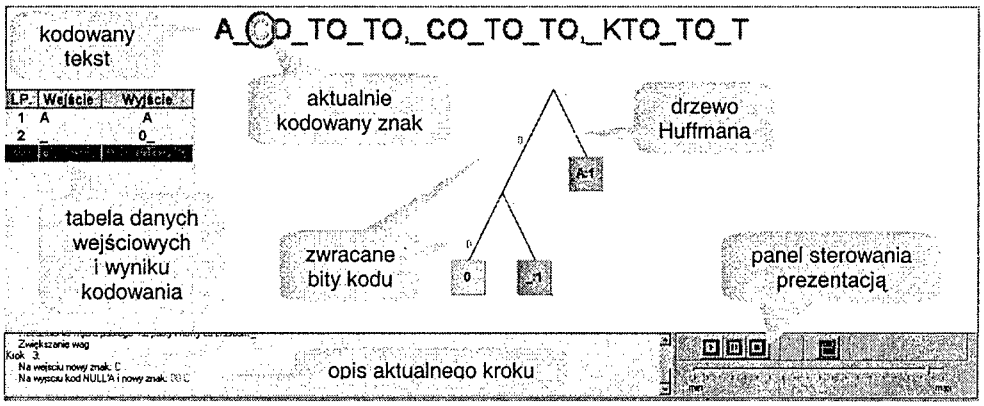
Rys. 2.1. Synteza kwasu DNA:
a) tworzenie się podwójnej helisy DNA, b) gotowa helisa DNA [10].

2.3. Dynamiczna wizualizacja parametru nie będącego czasem

Poznanie kształtu skomplikowanej bryły (np. cząsteczki jakiegoś złożonego związku chemicznego) wymaga zwykle obejrzenia jej z wielu stron. Efekt taki można uzyskać za pomocą animacji ukazującej bryłę (cząsteczkę) z punktu widzenia poruszającego się wokół obiektu obserwatora. Jest to chyba najpopularniejszy przykład wizualizacji dynamicznej nie związanej z czasem. Innym przykładem tego typu wizualizacji jest uszczegółowienie informacji polegające na prezentacji sekwencji obrazów przedstawiających przejście od jednego modelu do innego (np. animacja modelu cząsteczki chemicznej ujawniająca wiązania atomowe, jej krańcowymi kadrami mogłyby być obie części rys. 1.2). Uszczegółowienie informacji najczęściej jednak polega na przejściu od modelu ogólnego do jego szczegółów na zasadzie zbliżenia (np. animacja, której krańcowymi kadrami są obie części rys. 2.2). Odmiennym przykładem dynamicznej wizualizacji parametru innego niż czas (choć w pewnej mierze związanego z czasem) jest wizualizacja algorytmów, gdzie do ukazania następstwa kroków wykorzystuje się animację (np. algorytmy kompresji [1] – rys. 2.3).



Rys. 2.2. Prędkość przepływu cieczy (wyrażona liczbą grotów na linii) przez lejek (rurę): a) cały lejek, b) węższa, wyjściowa część lejka [10].



Rys. 2.3. Kadr z programu prezentującego statyczną metodę Huffmana kompresji danych [1].

2.4. Wizualizacja dynamiczna

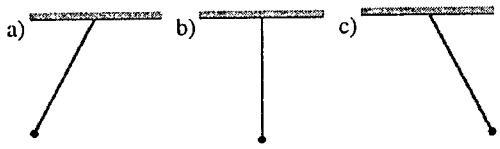
Animacja pokazująca upływ czasu stosowana jest głównie do pokazania ruchu [5]. Okazuje się, że nawet zaskakująco proste modele wyglądają bardzo przekonywująco, jeżeli tylko poruszają się w realistyczny sposób [2, 3]. Krążenie Księżyca wokół Ziemi (rys. 2.4) czy oscylacje wahadła matematycznego (rys. 2.5) przekonują o naturalności takiego typu prezentacji [10]. Co więcej, pierwszy z tych przykładów pokazuje, że prędkość animacji nie musi (w tym przypadku nawet nie powinna) być zgodna z czasem rzeczywistym.



Rys. 2.4. Obroty Księżyca wokół Ziemi (kadr z animacji) [10].

Wizualizacja dynamiczna jest wykorzystywana także w systemach czasu rzeczywistego, gdzie czas uzyskania wyniku jest bardzo istotny. Otrzymanie wyniku samo w sobie przestało mieć tak ogromne znaczenie, jak otrzymanie go we właściwym czasie. Po przekroczeniu ustalonego czasu wyniki mogą przestać być ważne. Wyróżnia się dwa rodzaje systemów czasu rzeczywistego – łagodny i rygorystyczny. W pierwszym z nich dopuszcza

się okazjonalne przekroczenie limitów czasu (np. wizualizacja informacji giełdowej), w drugim nie może wystąpić ani jedno przekroczenie, gdyż może to spowodować bezpośrednie zagrożenia życia ludzkiego (np. wizualizacja informacji o stanie reaktora jądrowego).



Rys. 2.5. Ruch wahadła matematycznego (kadry z animacji) [10].

3. WIZUALIZACJA DANYCH

Wizualizacja danych polega na przedstawieniu zależności między wartościami parametrów charakteryzujących badane zjawisko. Często zależności te dotyczą wzajemnych powiązań między wieloma atrybutami. Długość drogi hamowania samochodu zależy – na przykład – od wielu czynników, takich jak prędkość, rodzaj nawierzchni, jej wilgotność, stan opon i hamulców, itd. Prędkość pojazdu wpływa z kolei nie tylko na długość drogi hamowania, ale również na zużycie paliwa, poziom hałasu, wibracje, itp. Zależności mogą być zatem wielowymiarowe, i to zarówno od strony wielkości wpływających, jak i wynikowych. W związku z tym wizualizację danych można rozpatrywać w dwóch aspektach:
- wielowymiarowość dziedziny zależności (przykład z długością drogi hamowania),
- wielowymiarowość przeciwdziedziny zależności (przykład z prędkością pojazdu).

Ogólną postać zależności możemy zapisać wzorem (symbol $\prod$ oznacza iloczyn kartezjański):

$$\prod_{i=1}^{n} X_i \rightarrow \prod_{j=1}^{m} Y_j \,, \tag{3.1}$$

gdzie n, m – wymiary odpowiednio dziedziny i przeciwdziedziny; a X_i, Y_j – jednowymiarowe przestrzenie danych składające się na dziedzinę i przeciwdziedzinę. Zależność tę w dalszej części będziemy zapisywać w formie uproszczonej:

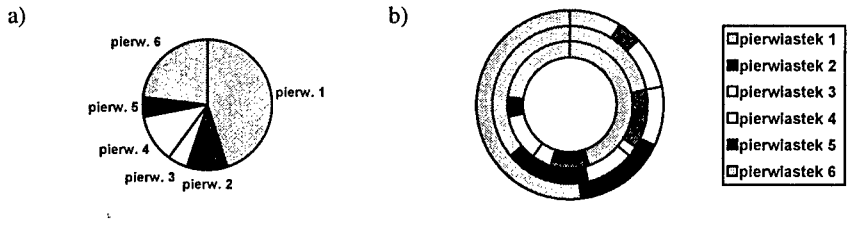
$$X^n \rightarrow Y^m. \tag{3.2}$$

Z najprostszą sytuacją mamy do czynienia oczywiście wówczas, gdy dziedzina i przeciwdziedzina są jednowymiarowe (rys. 1.3). Jeśli jednak jeden lub oba z wymiarów n lub m są duże, wówczas wizualizacja danych musi radzić sobie z przedstawianiem informacji wielowymiarowych przy wykorzystaniu w zasadzie jedynie 2 lub 3 wymiarów.

Poszczególne przestrzenie składowe dziedziny i przeciwdziedziny mogą mieć różny charakter: ciągły liczbowy (wielkości opisywane liczbami rzeczywistymi lub wymiernymi, np. czas, temperatura), dyskretny uporządkowany (wielkości dające się ponumerować i cechujące się uporządkowaniem, np. liczba elementów, ocena studencka, dzień tygodnia) bądź dyskretny nieuporządkowany (wielkości określające kategorie, gdzie kolejność nie ma znaczenia, np. marka samochodu, gatunek drzewa, stop metalu). Ten charakter danych powinien determinować rodzaj wykresu zastosowanego w wizualizacji. I tak na przykład wykres liniowy jest bez wątpienia lepszym sposobem wizualizacji zależności wielkości ciągłych liczbowych niż wykres słupkowy (rys. 1.3).

3.1. Wizualizacja zależności $X \rightarrow Y$

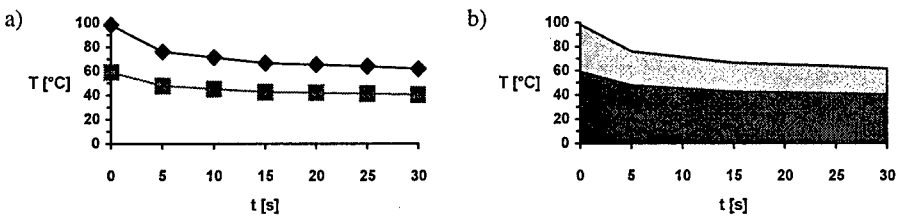
Do wizualizacji najprostszego rodzaju zależności, jakimi są zależności o dziedzinie i przeciwdziedzinie jednowymiarowej, doskonale nadają się tradycyjne sposoby obrazowania wykorzystujące dwuwymiarowy, najczęściej kartezjański układ współrzędnych. W przypadku dziedziny ciągłej najpopularniejszym sposobem prezentacji zależności są wykresy liniowe (rys. 1.3 a). Czasami stosuje się tutaj również wykresy punktowe, zwłaszcza przy danych uzyskanych wyrywkowo w wyniku pomiarów (rys. 1.3 a bez górnej linii). Wykresy punktowe wykorzystuje się również dla dziedziny dyskretnej. Tu jednak lepszą metodą pokazu wydają się wykresy słupkowe (rys. 1.3 b). Gdy przeciwdziedzina jest określona procentowo, to dla dziedziny dyskretnej można użyć wykresów kołowych (rys. 3.1 a).



Rys. 3.1. Zawartość pierwiastków w pewnych stopach metali:
a) wykres kołowy dla jednego stopu, b) wykres pierścieniowy dla trzech stopów.

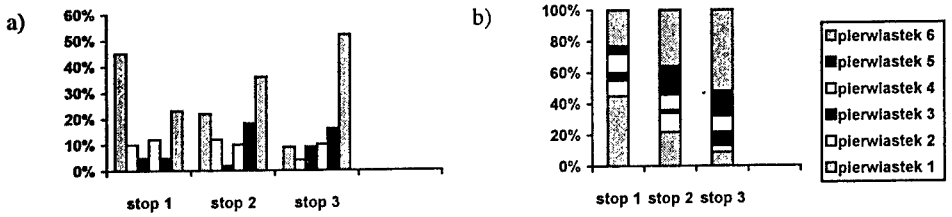
3.2. Wizualizacja zależności $X \rightarrow Y^m$ dla $m > 1$

Do wizualizacji zależności o dziedzinie jednowymiarowej i przeciwdziedzinie wielowymiarowej można w zasadzie adaptować techniki stosowane dla zależności $X \rightarrow Y$ poprzez zwielokrotnienie. Dla dziedziny ciągłej osiągnąć to można przez nałożenie na siebie wykresów punktowych lub liniowych (rys. 3.2 a). Poszczególne zależności „jednowymiarowe" na takim wykresie zbiorczym powinny być jednak łatwo rozróżnialne. W tym celu przebiegi zależności mogą być narysowane innymi kolorami bądź też z zastosowaniem różnych grubości czy stylów linii (ciągła, przerywana, kropkowana) lub odmiennych kształtów punktów (kółka, krzyżyki, trójkąty). Nieco inaczej uzyskuje się rozróżnienie zależności „jednowymiarowych" na wykresach warstwowych, gdzie poszczególne przebiegi zależności są uzupełnione polem poniżej przebiegu, sprawiając wrażenie nałożonych na siebie warstw malarskich (rys. 3.2 b).

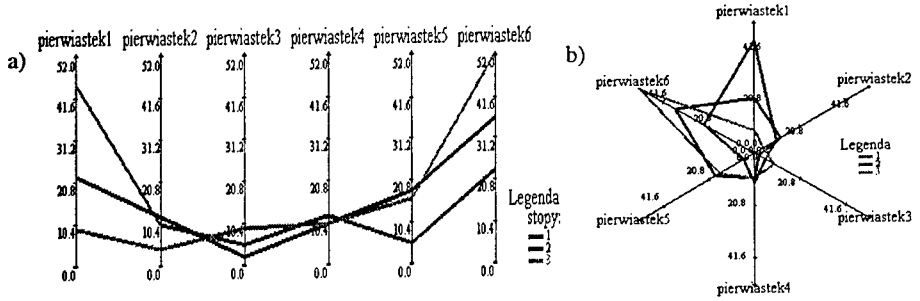


Rys. 3.2. Zależność temperatury wskazywanej przez wyjęty z wrzątku i z ciepłej wody termometr rtęciowy od czasu jaki upłynął od wyjęcia: a) wykres liniowy, b) wykres warstwowy.

Podobnie w przypadku dziedziny dyskretnej wykresy słupkowe o pojedynczych słupkach można łączyć w grupowane lub skumulowane wykresy słupkowe (rys. 3.3). Przykładem dostosowania wykresów kołowych do wielowymiarowej przeciwdziedziny są z kolei wykresy pierścieniowe (rys. 3.1 b). Zależności $X \rightarrow Y^m$ z dziedziną dyskretną można jednak przedstawiać za pomocą odmiennych metod wizualizacji, gdzie w miejsce linii pionowych dziedzinę reprezentuje zbiór łamanych, a osie przeciwdziedziny przyjmują postać prostych równoległych albo też pęku półprostych (rys. 3.4). Wykresy takie noszą miano odpowiednio wykresów typu „sieć rybacka" i wykresów radarowych (choć w drugim przypadku lepsze chyba by było określenie „sieć pajęcza").



Rys. 3.3. Zawartość pierwiastków w pewnych stopach metali:
a) grupowany wykres słupkowy, b) skumulowany wykres słupkowy.

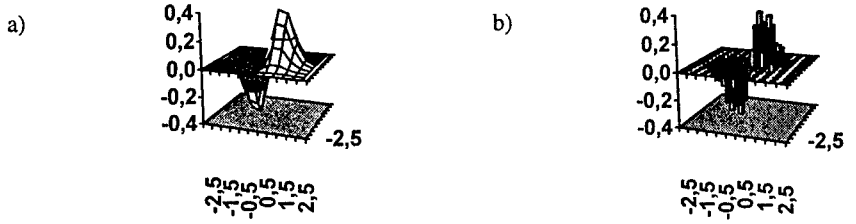


Rys. 3.4. Zawartość pierwiastków w pewnych stopach metali:
a) wykres „sieć rybacka", b) wykres radarowy [10].

3.3. Wizualizacja zależności $X^2 \rightarrow Y$

Również do wizualizacji zależności o dziedzinie dwuwymiarowej i przeciwdziedzinie jednowymiarowej można zaadaptować techniki stosowane dla zależności $X \rightarrow Y$ tym razem poprzez uprzestrzennienie czyli dodanie głębokości do wykresu, a więc wykorzystując trójwymiarowy, zwykle kartezjański układ współrzędnych. W ten sposób dla dziedziny ciągłej zamiast wykresów liniowych otrzymujemy wykresy powierzchniowe, a dla dziedziny dyskretnej – przestrzenne wykresy słupkowe (rys. 3.5).

Zależności $X^2 \rightarrow Y$ mogą być też przedstawiane jedynie w dwóch wymiarach geometrycznych – na płaszczyźnie odpowiadającej dziedzinie. Wówczas dla dziedziny ciągłej przeciwdziedzina może być reprezentowana barwą lub liniami wyrażającymi te same wartości (izolinie) [8]. W ten sposób od lat obrazowana jest wysokość na tradycyjnych mapach fizycznych stosowanych w geografii. W przypadku dziedziny dyskretnej wartości przeciwdziedziny mogą zaś być wyrażane za pomocą rozmiaru zadanych obiektów geometrycznych, np. kół na wykresach bąbelkowych.



Rys. 3.5. Funkcja $f(x,y) = x \cdot \exp(-x^2-y^2)$:
a) wykres powierzchniowy, b) przestrzenny wykres słupkowy.

3.4. Wizualizacja zależności $X^2 \rightarrow Y^m$ dla $m > 1$

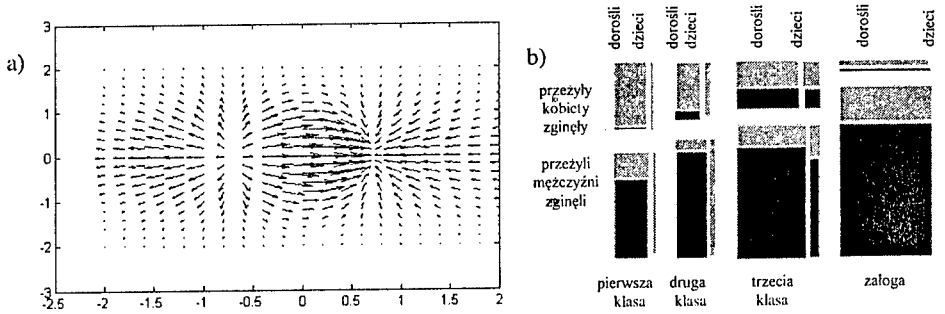
Wizualizacja zależności o dziedzinie dwuwymiarowej i przeciwdziedzinie wielowymiarowej wykorzystuje najczęściej jedynie dwa wymiary geometryczne – płaszczyznę odpowiadającą dziedzinie. Wtedy dla dziedziny ciągłej poszczególne wymiary przeciwdziedziny reprezentowane są najczęściej odpowiednio składowymi barwy (np. odcień, nasycenie, jasność), kreskowaniami (teksturą), izoliniami, itp. Dyskretne składowe przeciwdziedziny powinny być obrazowane za pomocą granic między różnymi wartościami, o ile tylko występują tendencje do występowania tych wartości w zwartych grupach (np. przynależność terytorialna na mapach geograficznych). Jeden z wymiarów przeciwdziedziny może być ponadto uzyskany przez dodanie trzeciego wymiaru geometrycznego. W przypadku dziedziny dyskretnej wartości przeciwdziedziny mogą zaś być wyrażane za pomocą rozmiaru zadanych obiektów geometrycznych, ich kształtu, barwy, itd.

Jeśli przeciwdziedzina jest tylko dwuwymiarowa i odpowiednio gładka (tzn. wartości zmieniają się dość wolno), to do jej wizualizacji można użyć metafory pola wektorowego [8]. Przeciwdziedzina przedstawiona może być wówczas za pomocą gęsto umieszczonych strzałek-wektorów (lub linii prądu wyznaczających tory cząstek w polu), których parametry (współrzędne lub długość i kierunek) reprezentują wartości przeciwdziedziny (rys. 3.6 a).

3.5. Wizualizacja zależności $X^n \rightarrow Y^m$ dla $n > 1$ i $m \geq 1$

Wizualizacja zależności o dziedzinie wielowymiarowej jest najtrudniejsza. Jeżeli jeden z wymiarów reprezentuje czas, to najlepszym sposobem wizualizacji jest animacja lub też sekwencja sąsiadujących ze sobą rysunków. Wówczas poszczególne kadry stanowią wizualizację zależności bez czasu $X^{n-1} \rightarrow Y^m$ w kolejnych chwilach czasu. Jeśli natomiast czas nie występuje w dziedzinie, podobny zabieg można wykonać na jednej ze składowych dziedziny uzyskując w konsekwencji albo dynamiczną wizualizację parametru nie będącego czasem, albo ciąg wykresów stanowiących swoiste przekroje przez zależność dla kolejnych wartości wybranej składowej dziedziny. Dla dziedziny trójwymiarowej sprowadza się to do wykonania sekwencji wykresów zależności o dziedzinie dwuwymiarowej $X^2 \rightarrow Y^m$. Przykładem takiej wizualizacji jest obraz mózgu w badaniu tomograficznym ($X^3 \rightarrow Y$).

W przypadku dziedziny ciągłej trudno podać ogólne zasady wizualizacji zależności $X^n \rightarrow Y^m$. Na szczęście dla dziedziny dyskretnej sytuacja jest prostsza. Mianowicie dla przeciwdziedziny jednowymiarowej ($m = 1$) stosować można wykresy mozaikowe, przedstawiające wyniki jako rozmiary prostokątów (rys. 3.6 b). Przy przeciwdziedzinie wielowymiarowej ($m > 1$) prostokąty te mogą być dodatkowo kolorowane [9]. W takim przypadku można też wykonać wykres mozaikowy dla każdego wymiaru przeciwdziedziny.



Rys. 3.6. Przykłady wizualizacji zależności wielowymiarowych:
a) pole wektorowe gradientu funkcji $f(x,y) = x \cdot \exp(-x^2-y^2)$ [10], b) wykres mozaikowy prezentujący liczby ocalonych (jaśniejsze prostokąty) i ofiar (czarne prostokąty) katastrofy Titanica [9, 10].

4. PODSUMOWANIE

W pracy omówiono różne technik wizualizacji informacji. Wyróżniono dwa podstawowe rodzaje wizualizacji: wizualizację naukową przedstawiającą wirtualny model zjawiska i wizualizację danych prezentującą zależności opisujące zjawisko. Przeanalizowano ponadto aspekt czasu w wizualizacji informacji i usystematyzowano metody wizualizacji danych ze względu na wymiar dziedziny i przeciwdziedziny zależności.

BIBLIOGRAFIA

[1] D. Boruchalska: *Wizualizacja przebiegu algorytmów bezstratnej kompresji danych w czasie rzeczywistym - praca dyplomowa.* Politechnika Gdańska WETI, Gdańsk 2003.
[2] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, R. L. Phillips: *Wprowadzenie do grafiki komputerowej.* WNT, Warszawa 1995.
[3] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes: *Computer Graphics: Principles and Practice, Second Edition.* Addison-Wesley, Reading 1990.
[4] M. Friendly: *Gallery of Data Visualization.* http://www.math.yorku.ca/SCS/Gallery/ 2000.
[5] R. S. Gallagher (ed.): *Computer Visualization.* CRC Press, Boca Raton 1995.
[6] V. Gillet: *Data Visualisation.* http://dagda.shef.ac.uk/inf308/material2003/DataVisualisation.pdf ↑ 2003.
[7] K. Hicks: *C. J. Minard's Map - Napoleon's March of 1812.* http://www.ddg.com/LIS/InfoDesignF96/Kelvin/Napoleon/map.html 1996.
[8] P. Krzyżanowski: *Wizualizacja w obliczeniach naukowych: standardowe techniki.* http://www.mimuw.edu.pl/~przykry/ON/index.html 2000.
[9] R. Spence: *Information Visualization.* ACM Press (+ Addison-Wesley), Harlow 2001.
[10] M. Stępniak: *Wielowymiarowa wizualizacja obliczeń w czasie rzeczywistym – praca dyplomowa.* Politechnika Gdańska WETI, Gdańsk 2003.

MULTIDIMENSIONAL VISUALIZATION OF INFORMATION

Summary

The paper systematizes methods for visual presentation of information. It contains description of different approaches applied in scientific and data visualization.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Marek Nahotko

Zakład Informacji Naukowej IINiB, Uniwersytet Jagielloński

ONTOLOGICZNE MODELOWANIE RZECZYWISTOŚCI W BIBLIOTEKACH CYFROWYCH

Streszczenie

W referacie przedstawiono podstawową ontologię i model logiczny ABC służący modelowaniu informacji w bibliotece cyfrowej. Przedstawiono słownik ABC i wyróżnione klasy. Wskazano na narzędzie ułatwiające współdziałanie ABC – tezaurus MetaNet

1. WSTĘP

Paul Ginsparg stwierdził [1], że *integracja informacji* stanowi podstawę dla bogatej „przestrzeni wiedzy" zbudowanej na wierzchołku podstawowej „warstwy danych" Web. Ten poziom wiedzy składa się z usług stwarzających wartość dodaną, które przetwarzają i oferują abstrakcyjną *informację* i *wiedzę*, a nie dostarczają *dokumentów* w sposób, do jakiego przyzwyczailiśmy się ze strony współczesnych wyszukiwarek Web.

Integracja informacji w Web realizowana jest przy pomocy wymiennych elementów składających się na architekturę będącą w centrum zainteresowań W3C i pokrewnych inicjatyw związanych z siecią semantyczną (ang. *Semantic Web*) [2]. Prace te dotyczą mechanizmów kodowania i manipulacji informacją (np. XML – http://www.w3.org/XML/, RDF [3], XSLT [4]) oraz konstrukcji ontologii i zasad wnioskowania (np. RDFS [5], DAML+OIL, OWL [6]). Integracja informacji inspiruje także w społeczności związanej z bibliotekami cyfrowymi wiele prac dotyczących metadanych; część z nich skupia się na problemach określonych zastosowań (np. FGDC – http://www.fgdc.gov, w zakresie przestrzeni geograficznej, IMS LTSC – http://www.imsproject.org/, w zakresie edukacji/kształcenia), inne inicjatywy metadanych wychodzą poza specyficzne problemy poszczególnych zastosowań w kierunku dostarczania usług w środowisku zasobów heterogenicznych (np. Dublin Core – http://dublincore.org/, służący przeszukiwaniu informacji w dowolnym zakresie) [7].

Istnieje jeszcze inny potencjalnie ważny element służący tworzeniu architektur dla integracji: *podstawowe ontologie* (ang. *core ontologies*) [8]. Ontologia jest tu rozumiana jako opis w języku formalnym jawnych i ukrytych pojęć i relacji odpowiedni dla wyrażania treści dokumentów w bibliotece cyfrowej. Celem tworzenia podstawowych ontologii jest realizacja globalnego i rozwojowego modelu, przy pomocy którego mogą być konwerto-

wane i integrowane dane pochodzące z różnych źródeł. Taka kanoniczna forma może następnie służyć jako wspólna podstawa wiedzy dla narzędzi operujących we wszystkich dziedzinach oraz dla usług takich jak przeszukiwanie informacji, przeglądanie i wyszukiwanie danych. Pełna i rozszerzalna ontologia wyrażająca podstawowe pojęcia wspólne dla różnych dziedzin może dostarczyć podwalin dla bardziej wyspecjalizowanych słowników i wykazów pojęć stosowanych w poszczególnych dziedzinach. Tego typu podstawowa ontologia ma zasadnicze znaczenie dla właściwego zdefiniowania konwersji pomiędzy reprezentacjami wiedzy specyficznymi dla poszczególnych dziedzin (tzn. słownikami metadanych) i dla późniejszej konstrukcji usług takich jak wyszukiwanie w wielu dziedzinach, przeglądanie, wyszukiwanie danych i wydobywanie wiedzy. Tworzenie tego typu usług wokół centralnego modelu pozwala na uniknięcie gwałtownego wzrostu ilości aplikacji oraz ich złożoności będącego oczywistym efektem wielokrotnej konwersji pomiędzy poszczególnymi formatami lub ontologiami.

Różnica pomiędzy podstawową ontologią a formatem *podstawowych metadanych*, takim jak Dublin Core, jest subtelna, lecz ważna. Oba mają na celu integrację informacji. Różni są ich adresaci. Metadane w zasadzie są tworzone, edytowane i przeglądane przez ludzi. Stąd też podczas ich projektowania zasadniczą rolę podczas ich projektowania powinien odgrywać czynnik ludzki, co prowadzi do koniecznych ograniczeń w ich złożoności. W przeciwieństwie do nich, podstawowa ontologia dostarcza podstawowego modelu formalnego dla narzędzi integrujących dane źródłowe i realizujących wiele rozbudowanych funkcji w sposób zautomatyzowany. W związku z tym możliwy do przyjęcia jest wyższy poziom złożoności, a proces jej projektowania służy nie tyle jej łatwemu rozumieniu, co kompletności i logicznej poprawności.

Podstawowa ontologia jest fundamentem dla planu integracji informacji opisanego przez Diego Calvanese i in. [9]. Głównym elementem tego planu jest pojęcie *modelu zadania* (ang. *enterprise model*) – „konceptualna reprezentacja globalnych pojęć i relacji stosowanych w aplikacji". Model zadania dostarcza podstaw do integracji otwartego zestawu *danych źródłowych* dostarczanych przez niezależne społeczności Web lub jakiekolwiek inne otwarte przestrzenie informacyjne. Konwersja możliwa jest dzięki:

- Utworzeniu *modeli źródłowych*, które pozwalają na konceptualną reprezentację każdej kategorii danych źródłowych. Modele te nie zawsze są kompletne (nie muszą reprezentować pełnej wiedzy w danych źródłowych), ale są wystarczające dla potrzeb danego zastosowania modelu zadania.
- Zdefiniowaniu *modelu domeny*, który wyraża współzależności wewnętrzne i zewnętrzne pomiędzy zadaniem i modelami domeny (dziedziny zastosowań).
- Formalizacji wymienionych modeli konceptualnych i relacji poprzez struktury umożliwiające podejmowanie decyzji takie jak *logika opisowa*.

Tego typu zintegrowany model może być użytkowany podczas świadczenia usług wielodziedzinowego wyszukiwania informacji. Jeżeli integracja informacji jest właściwa, to zapytanie do *modelu zadania* powinno spowodować wydanie takiej samej odpowiedzi, jaka zostałaby przekazana w wyniku wyszukiwania wg tego samego zapytania we wszystkich źródłach. Model ten powinien umożliwić agregację nie tylko informacji ekwiwalentnej (np. wszystkie publikacje autora w określonym okresie), ale także informacji komplementarnej – np. opis spotkań Van Gogha z Gauguinem wraz z przedstawieniami ich prac z tego okresu.

Po to, aby dostarczyć podstaw dla tego typu modelu zadania, dobrze sformułowana podstawowa ontologia powinna być *kompletna* i *rozszerzalna*, wyrażając podstawowe pojęcia wspólne dla wielu różnych dziedzin i dostarczając podstawy dla specjalizacji

w zakresie szczegółowych pojęć i słowników dla poszczególnych dziedzin. Ontologia podstawowa oprócz wprowadzania typologii powinna także dostarczać wspólnego języka służącego do definiowania relacji pomiędzy elementami informacji. Według M. Doerra [10], podstawowa ontologia powinna być „ukierunkowana na cechy" – dostarczając wspólnego języka definiującego sposób powiązania relacjami elementów informacji.

2. ABC – ONTOLOGIA DLA MODELOWANIA INFORMACJI

Jednym ze sposobów tworzenia podstawowej ontologii jest przygotowanie „reprezentacji całej wiedzy" – ten ambitny cel postawiły przed sobą takie inicjatywy jak Cyc (http://www.cyc.com./). Przydatność i możliwości zastosowania tego rodzaju podejścia całościowego mogą być potwierdzone tylko w długim okresie czasu. Na bieżąco można zastosować jednak bardziej praktyczne podejście polegające na wykorzystaniu i harmonizacji wiedzy przedstawicieli głównych społeczności użytkowników informacji.

W tej części referatu przedstawione zostaną prace społeczności związanej z ontologią ABC [11], będącej narzędziem służącym tworzeniu podstawowej ontologii dla informacji w bibliotece cyfrowej. Ontologia ABC odzwierciedlają próby wyrażenia wspólnych pojęć będących podstawą struktur danych wykorzystywanych przez społeczności związane z wymianą informacji.

2.1. Ontologia ABC

Ontologia ABC powstała w odpowiedzi na potrzebę integracji w bibliotekach cyfrowych informacji pochodzącej z wielu źródeł. Jest ona produktem Projektu Harmony (http://metadata.net/harmony/).

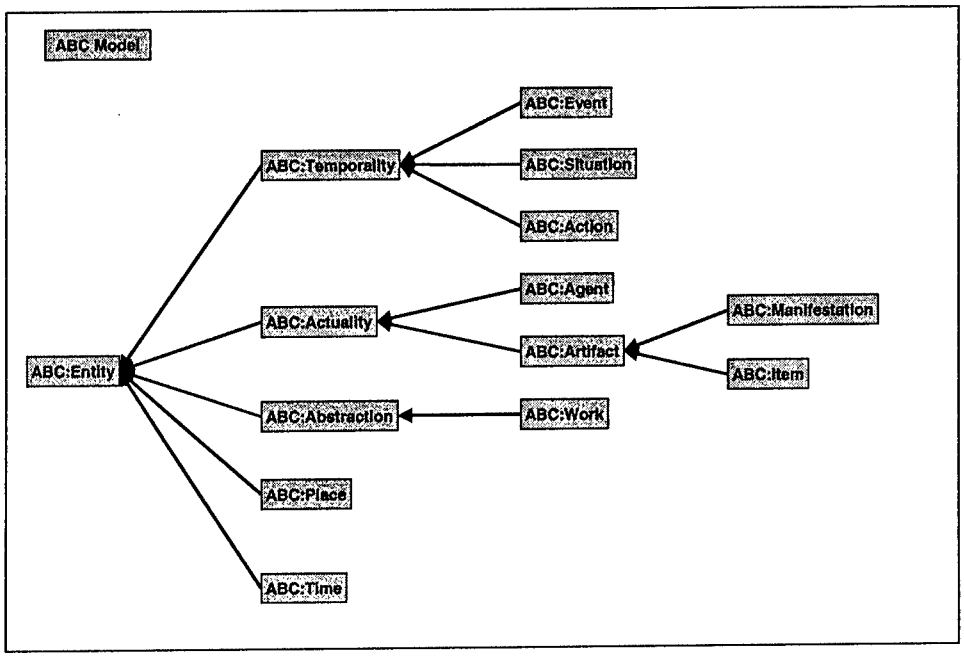
Specjaliści zatrudnieni do prac nad Projektem Harmony byli wcześniej związani z realizacją innych projektów i inicjatyw dotyczących metadanych, takich jak Dublin Core i MPEG-7. Na podstawie tych doświadczeń wyciągnięto kilka zasadniczych wniosków:

- ilość słowników metadanych będzie nadal rosła ze względu na potrzebę tworzenia struktur informacji gromadzonej dla celów lokalnych;
- próby utworzenia uniwersalnych słowników metadanych są chybione mniej więcej z tych samych przyczyn, dla których języki „naturalne" (używane przez społeczności do bezpośredniego opisu treści) w nieunikniony sposób różnicują się (historia jest pełna nieudanych prób stworzenia wspólnych języków naturalnych);
- bardziej przydatne są prace nad sformułowaniem podstawowego języka „porozumiewawczego" lub „przetwarzania", podstawowej ontologii, przez włączenie podstawowych encji i relacji wspólnych dla różnych słowników metadanych;
- tego typu ontologia może być użyteczna dla wielu celów, takich jak integracja informacji z różnych słowników oraz dostarczanie podstawowych pojęć, które przyszłe inicjatywy metadanych mogą wykorzystać do budowy własnych słowników, specyficznych dla danego zastosowania.

ABC opisuje aspekty czasowe przy pomocy encji modelujących pierwszej klasy. Jak stwierdzono w [12], modelowanie zmian w czasie ma zasadnicze znaczenie dla opisu treści cyfrowych ze względu na zmienność będącą ich cechą immanentną, a także ze względu na ich niepewną proweniencję lub rzetelność. ABC zawiera notacje dotyczące zarówno „wydarzeń" jak i „sytuacji", które odpowiednio modelują procesy i cechy egzystencjalne. Inspiracją tych koncepcji są modele takie jak Sieci Petriego [13] i rozszerzenia czasowe dla logiki pierwszego rzędu takie jak Rachunek Sytuacyjny [14]. Dzięki temu ABC zdolne jest

modelować okresy czasu, podczas gdy inne cechy obiektu pozostają statyczne. Możliwe jest także modelowanie wydarzeń lub przejść związanych z modyfikacją cech, np. gdy zmienia się wersja obiektu.

ABC zbudowane jest zgodnie z koncepcjami zawartymi w modelu FRBR [15]. Pojęcia tam zawarte – *utwór, uzewnętrznienie, materializacja, pozycje* – pozwalają ABC na tworzenie odnośników do jednostek, które łączy wspólne pochodzenie praw własności intelektualnej. Chociaż model ABC nie jest zależny od RDF ani RDFS, to podstawowe zasady modelu danych RDF silnie na niego wpłynęły – chodzi głównie o definicje klas i podklas dla encji oraz cech i pod-cech dla relacji pomiędzy tymi encjami.



Rys. 1: Pełna hierarchia "IsA" klas ABC [19]

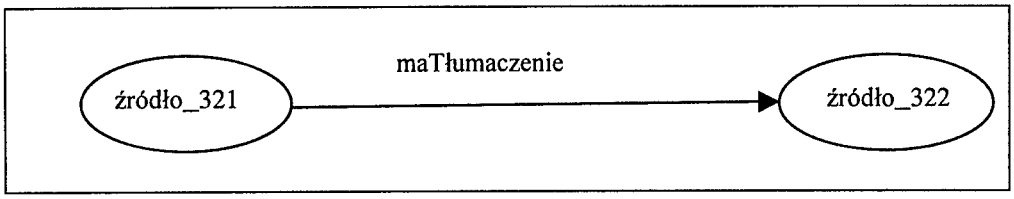
2.2. Model ABC dla współdziałania metadanych

Model logiczny ABC [16] jest zbudowany na bazie wielu własnych koncepcji oraz założeń, w tym powszechnej identyfikowalności źródeł, własności (jako specjalnego typu źródła) oraz klas, które zawierają zestawy źródeł (i własności). Model ten definiuje także zestaw podstawowych klas (zestawów źródeł), takich jak wytwory, wydarzenia, czynnik aktywny i relacje. Poprzez dzielenie podstawowych klas na podklasy powstają słowniki metadanych specyficzne dla określonej aplikacji lub określonej dziedziny.

Źródła powiązane są relacjami w różny sposób: najczęstszymi są zawieranie się, tłumaczenie i pochodzenie. Opis tych relacji jest istotnym aspektem metadanych. W pewnych systemach (np. Dublin Core) opisy relacji są dość proste; w innych istnieje potrzeba bardziej szczegółowego opisu. Model ABC, dzięki wykorzystaniu modelu danych grafu RDF, pozwala na przechodzenie pomiędzy złożonymi i prostymi opisami relacji. W utworzonym modelu jednostka wejściowa dla relacji i jednostka wyjściowa dla relacji oraz relacja pomiędzy tymi dwoma jednostkami są uważane za źródła. W celu wzbogacenia

opisu tych źródeł możliwe jest następnie dołączenie do nich informacji o ich właściwościach. W ten sposób połączono prostą relacją źródło pierwszej klasy ze związanymi z nim właściwościami.

Przykład na rys. 2. ilustruje sytuację dotyczącą relacji „maTłumaczenie". W prosty sposób możemy stwierdzić, że jakiś dokument jest obcojęzyczną wersją innego dokumentu.



Rys. 2. Prosta relacja między źródłami

Takie podejście ma zastosowanie do wydarzeń związanych z wejściem i wyjściem źródeł lub opisujących wpływ czynnika działającego na źródło. Przykładami takich par relacji wydarzenie/relacja są:

Wydarzenie: Modyfikacja ⇔ relacja *JestWersją*
Wydarzenie: Kompilacja ⇔ relacja *SkompilowaneZ*
Wydarzenie: Wydobycie ⇔ relacja *WydobyteZ*
Wydarzenie: Przeformatowanie ⇔ relacja *JestFormatem*
Wydarzenie: Tłumaczenie ⇔ relacja *TłumaczenieZ*
Wydarzenie: Pochodzenie ⇔ relacja *PochodziZ*

Do modelowania wydarzeń wykorzystuje się **słownik ABC**. Słownik ABC definiuje i określa podstawowy zestaw głównych klas pojęć abstrakcyjnych, wspólnych dla wszystkich użytkowników metadanych. Główne klasy służą dla stworzenia punktów dostępu dla różnych własności (lub metadanych) związanych z treścią informacji i jej funkcjonowaniem. Będą one stanowić podstawową infrastrukturę dla modelowania metadanych i dla uszczegółowienia przez tworzenie podklas.

Przegląd różnych modeli metadanych[1] [17] pozwala na wyróżnienie następujących jednostek wspólnych:

Źródła
Wydarzenia
Wejścia i Wyjścia
Działanie
Kontekst (zawierający Czas i Miejsce)
Relacje wydarzenia

Źródła są superklasą zawierającą wszystkie możliwe rzeczy zawarte w uniwersum – mogą one być fizyczne, cyfrowe lub abstrakcyjne. Każde źródło ma odpowiedni, unikalny identyfikator.

Wydarzenia są działaniem lub zjawiskiem. Każde wydarzenie ma swój Kontekst (Czas i/lub Miejsce) z nim związany (chociaż nie musi być on wyrażony wprost). Wydarzenia mogą także mieć związane ze sobą **wejścia** i **wyjścia**. Np. wydarzenia generujące nowe lub zmodyfikowane źródło (np. tłumaczenie, modyfikacja) będą miały zarówno wejścia jak i wyjścia.

[1] Uwzględniono takie modele metadanych jak: IFLA FRBR, CIDOC, INDECS, MPEG-7 i Dublin Core.

Działanie jest dodatkiem do wydarzenia, które jest wykonywane przez jednego lub więcej aktorów lub czynników działających (agentów) odgrywających określone role. Działanie może istnieć jedynie jako cecha Wydarzenia. Każde Działanie posiada jedną lub więcej cech Agenta i opcjonalną cechę Roli.

Czynnik działający (Agent) reprezentuje źródło, które działa podczas wydarzenia – lub „aktorów" wydarzenia. Agent jest cechą Działania i zwykle posiada związaną Rolę, która definiuje rolę, jaką aktor odgrywa w określonym wydarzeniu. Można wyróżnić najczęściej używane rodzaje agentów:

osoba/istota ludzka;

organizacja;

instrument (sprzęt komputerowy, oprogramowanie, maszyna).

Każde źródło może odegrać rolę sprawczą, co pozwala mu działać jako agentowi. Pozostałe możliwe rodzaje agentów obejmują: zwierzęta, zwierzęta fikcyjne (np. Teletubbies), istoty pozaziemskie, istoty ponadnaturalne, stwory wyimaginowane, obiekty puszczone w ruch (np. obraz, który spadł ze ściany i uderzył w rzeźbę, która się rozbiła i następnie jest prezentowana jako nowe źródło w muzeum), procesy naturalne lub występujące w środowisku (burze, plagi, erozja, rozkład itp.).

Kontekst:

Data/Czas. Czas może być wyrażony w różny sposób. Może to być zarówno swobodny tekst opisujący okres lub wydarzenie lub też dane o dacie/czasie w określonym formacie. Może to być zarówno czas bieżący lub okres czasu. Może to być GMT, czas lokalny lub czas odniesiony do zakresu określonego obiektu, np. znak czasu na taśmie wideo.

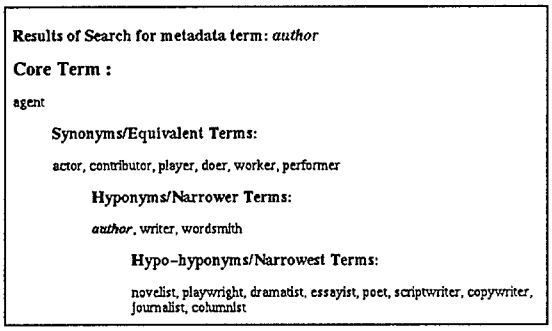
Miejsce. Miejsce opisuje lokalizację przestrzenną. Może być ono podane w formie swobodnego tekstu lub w bardziej sformatowany sposób. Może ono być absolutne lub posiadać odniesienia. Może to być punkt, linia, obszar dwu- lub trójwymiarowy. Podobnie jak czas, miejsce może być wyrażane bardzo różnie. Może to być lokalizacja przestrzenna dotycząca świata rzeczywistego lub lokalizacja związana z określonym pochodzeniem, systemem współrzędnych lub rozmiarów obiektów.

Relacje zostały wprowadzone dla wyrażania związków pomiędzy wydarzeniami. Można wyróżnić relacje najwyższego poziomu takie jak: czasowe, przestrzenne, czasoprzestrzenne, nieobowiązkowe, warunkowe. Każda z nich może mieć wiele podrodzajów, np. relacje czasowe mogą zawierać: poprzedni, jednoczesny, równy, zawarty, następny. Relacje mogą także posiadać atrybuty kierunku (jednokierunkowe i dwukierunkowe) i stopnia (pojedynczy, podwójny, n-stopnia) z nimi związane. Relacje warunkowe posiadają określenia jednego lub wielu warunków z nimi związanych.

Autorzy modelu ABC zaproponowali specjalny język służący przenoszeniu elementów metadanych pomiędzy dużą ilością systemów semantycznych, specyficznych dla określonych zastosowań. Podstawą tego języka jest tezaurus MetaNet [18]. Jego celem jest udostępnienie wiedzy semantycznej niezbędnej dla umożliwienia maszynowego odczytywania relacji synonimii i hierarchii pomiędzy terminami metadanych pochodzącymi z różnych dziedzin zastosowań. Terminologia tezaurusa ograniczona jest do najważniejszych modeli i słowników metadanych służących opisowi atrybutów i wydarzeń związanych z obiektami informacyjnymi. Dotyczy to słowników metadanych wykonanych w środowiskach bibliotek, muzeów, archiwów i użytkowników praw autorskich. Dla celów MetaNet przygotowano czytelną maszynowo reprezentację schematu w RDF. Dla określenia relacji hierarchicznych i relacji typu obiekt/atrybut wykorzystywane są elementy RDF *Class*, *subClassOf*, *property*, *subPropertyOf*. Podstawowy słownik ABC służy jako

zestaw terminów preferowanych najwyższego poziomu. Pomimo tego, że słownik ten utworzony został metodą manualną, można łatwo sobie wyobrazić jego automatyczne wygenerowanie przy użyciu mechanizmów wnioskowania służących łączeniu schematów RDF z różnych dziedzin.

Rozważmy np. termin podstawowy ze słownika ABC „Agent", który jest preferowanym terminem w tezaurusie MetaNet. Semantycznie równoważnymi terminami dla „Agent" często używanymi w innych słownikach metadanych są: *actor, contributor, creator, player, doer, worker, performer*. Węższymi terminami lub hiponimami dla „Agent" są: *author, composer, artist, musician* itd.



Rys. 3. Wynik wyszukiwania w MetaNet

Utworzono także interfejs umożliwiający przeglądanie i wyszukiwanie w MetaNet w środowisku Web. Użytkownicy mogą wyszukać dowolny termin metadanych i przeglądać listę terminów powiązanych z nim relacjami synonimii i hierarchii. Na rys. 3 przedstawiono wynik wyszukiwania według terminu „author".

3. ZAKOŃCZENIE

Słownik ABC nie jest jedynie słownikiem metadanych. Jest on głównie podstawowym modelem i ontologią pozwalającą na tworzenie słowników specyficznych dla danej dziedziny lub społeczności. Ontologia ABC zawiera wiele podstawowych encji i relacji wspólnych dla wielu słowników metadanych. Społeczności planujące utworzenie własnego słownika metadanych mogą więc uczynić to poprzez odpowiednie rozszerzenie modelu ABC.

Model ABC powinien pozwolić na stworzenie właściwego schematu podstawowego służącego modelowaniu, walidacji, przechowywaniu, nawigacji i wyszukiwaniu metadanych różnego rodzaju, tworzonych przez wiele różnorakich narzędzi.

W 2001 roku powstała grupa robocza o nazwie DELOS, której zadaniem jest harmonizacji ontologii ABC i CIDOC/CRM (CIDOC object-oriented Conceptual Reference Model), ontologii utworzonej na potrzeby muzealnictwa. Zadaniem tej grupy roboczej jest połączenie ontologii ABC i CIDOC CRM w pojedynczą ontologię. W efekcie niezbędna jest realizacja takich prac jak:

- przygotowanie metodologii służącej porównywaniu, łączeniu i rozpowszechnianiu ontologii;
- badanie alternatywnych możliwości reprezentacji ontologii;
- realizacja optymalnego sposobu zarządzania ontologiami scalonymi, z zapewnieniem możliwości dołączania kolejnych ontologii.

Efekty tych prac przedstawione są w [19].

BIBLIOGRAFIA

[1] Ginsparg P.: *Creating a global knowledge network*, W: Proceedings of the Second ICSU-UNESCO Conference on Electronic Publishing in Science. February 20-23 2001, Paris, France. Dostęp: http://users.ax.ac.uk/~icsuinfo/ginspargfin.htm

[2] Berners-Lee T., Hendler J., Lassila O.: *The Semantic Web*. Scientific American 2001 May, Vol 50 http://www.sciam.com/article.cfm?chanID=sa006&colID=1&articleID=00048144-10D2-1C70-84A9809EC588EF21

[3] Clyne G., Carroll J.: *Resource Description Framework (RDF):Concepts and Abstract Syntax*.W3C Proposed Recommendation, 15 December 2003

[4] Clark J.: *XSL Transformations (XSLT)*W3C Recommendation, 16 November 1999

[5] Brickley D., Guha R.V: *RDF Vocabulary Description Language 1.0: RDF Schema.* W3C Proposed Recommendation15 December 2003

[6] McGuinness D., van Harmelen F.: *OWL Web Ontology Language Overview*.W3C Proposed Recommendation 15 December 2003

[7] Nahotko M.: *Metadane jako narzędzie opracowania elektronicznych zasobów sieciowych.* Praktyka i Teoria Informacji Naukowej i Technicznej 2003 Nr 4 s. 44-49

[8] Nahotko M.: *Semantyczny Web i jego ontologie.* EBIB Elektroniczny Biuletyn Informacyjny Bibliotekarzy 2003 Nr 10 (49)

[9] Calvanese D. [i in.]: *Description Logic Framework for Information Integration.* W: Sixth International Conferenceon Principles of Knowledge Representation and Reasoning (KR'98). June, 2-5 1998, Trento, Italy

[10] Doerr M.: *The CIDOC/CRM – an ontology Approach to Semantic Interoperability of Metadata.* AI Magazine – Special Issue on Ontologies, 2002 Vol 23 Nr 4

[11] Lagoze C., Hunter J.: *The ABC Ontology and Model.* Journal of Digital Information 2001 Vol. 2, Nr 2 http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Lagoze/

[12] Lagoze C.: *Bussiness Unusual: How "Event Awareness" May Brethe Life into the Catalog.* W: Bicentennial Conference on Bibliographic Control in the New Millennium. November 15-17 2000, Washington DC, USA http://lcweb.loc.gov/catdir/bibcontrol/lagoze.html

[13] Peterson J.L.: *Petri net theory and the modeling of systems.* Englewood Cliffs N.J.: Prentice-Hall 1981

[14] McCarthy J.: *Programs with common sense.* W: M. Minsky [red.] Semantic Information Processing. Cambridge, MA MIT Press 1968, s. 403-418

[15] Nahotko M.: *Functional Requirements for Bibliographic Record (FRBR) - model opracowania zbiorów bibliotecznych IFLA.* Bibliotekarz 2001, nr 1, s. 13-16

[16]Brickley D., Hunter J., Lagoze C.: *ABC: a Logical Model for Metadata Interoperability.* http://www.ilrt.bris.ac.uk/discovery/harmony/docs/abc/abc_draft.html

[17] Lagoze C., Hunter J., Brickley D.: *An Event-Avare Model for Metadata Imteroperability.* W: Research and Advanced Technology for Digital Libraries. 4th European Conference ECDL 2000 Proceedings. September 2000. Lisbon, Portugal. Berlin: Springer 2000, s. 103-116

[18] Hunter J.: *MetaNet – A Metada Term Thesaurus to Semantic Interoperability Between Metadata.* Journal of Digital Information 2001 Vol. 1 Nr 8 http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Hunter/

[19] Doerr M. [i in.]: *Building Core Ontologies. A White Paper of the DELOS Working Group on Ontology Harmonization.*

ONTOLOGIC MODELS OF REALITY IN DIGITAL LIBRARIES

Summary

In the paper it is described core ontology and logic model named ABC. It is used for information modelling in digital library. Vocabulary of ABC and clases are presented. The tool facilitating interoperability of ACB – MetaNet thesaurus is introduced.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Marcin Pazio

Katedra Systemów automatyki, Politechnika Gdańska

ZASTOSOWANIE ZACHOWUJĄCEJ KOLOR SEGMENTACJI OBRAZU BARWNEGO DO WYSZUKIWANIA OBSZARÓW ZAWIERAJĄCYCH TEKST

Streszczenie

W niniejszym artykule opisana została metoda segmentacji obrazów barwnych dedykowana do rozpoznawania obszarów zawierających tekst oraz przedstawiona została metoda kontekstowej analizy przetworzonego obrazu.

1. WSTĘP

Poruszając się dużą część informacji na temat otaczającego nas świata czerpiemy z obrazu. Odbierane przez nas obrazy zawierają także informację tekstową. Wyodrębnienie z obrazu tekstu i zamiana go za pomocą narzędzi OCR (optical character recognition) z postaci graficznej na postać tekstową umożliwia z kolei zamianę tekstu na syntetyczny tekst mówiony. Taka zaś postać tekstu może być bardzo pomocna dla osób niewidomych i niedowidzących.

Narzędzia do syntezy głosu oraz programy OCR są obecnie dostępne. Niestety, znane programy OCR nie radzą sobie z tekstami znajdującymi się w obrazach pozyskanych z otaczającego nas świata. Wymagają one z reguły dobrze zdefiniowanego tekstu, umieszczonego na jednolitym tle, umieszczonego na równoległych liniach. Należy zatem stworzyć metodę optymalizującą obraz i izolującą tekst z obrazu oryginalnego przed jego analizą OCR.

Niniejszy artykuł opisuje segmentację obrazu oraz prostą metodę kontekstowej analizy segmentów, pozwalającą zarówno na wyodrębnienie ciągów segmentów mogących być tekstem, jak i określenie podstawowych parametrów tekstu tworzonego przez te segmenty: wysokości liter, kolor, lokalizacja w obrazie.

2. SEGMENTACJA OBRAZU

Algorytm segmentacji powinien uwzględniać charakter przetwarzanego obrazu i cel przetwarzania, cechy szczególne poszukiwanych obszarów obrazu i działać całkowicie automatycznie.

2.1. Źródło i parametry obrazu

Obraz poddawany analizie może pochodzić praktycznie wyłącznie z kamery lub cyfrowego aparatu fotograficznego. Dostępne rozwiązania przetworników obrazu tych urządzeń opierają się na analizie obrazu w przestrzeni RGB. Istnieją dwa główne rozwiązania:

1 Obraz jest przed analizą rozłożony na składowe RGB przez system filtrów i analizowany oddzielnie dla każdej składowej przez oddzielny przetwornik obrazu.
2 Przetwornik wyposażony jest w filtr mozaikowy. Analiza kolorystyczna dokonywana jest przez jeden przetwornik, a pełna informacja na o wszystkich składowych barwy wymaga uzupełnienia na podstawie informacji z pikseli sąsiadujących.

Drugie z rozwiązań jest częściej stosowane, ale uzyskiwany na wyjściu obraz posiada znacznie mniejszą ilość informacji opisujących barwę niż jasność. Efektywna rozdzielczość dla informacji o barwie jest o ok. 2/3 mniejsza. To znacznie utrudnia segmentację gdy kryterium utworzenia segmentu z obszaru związane jest jedynie z występowaniem zmian parametrów chromatycznych. W takim przypadku granice obszarów są mocno rozmyte, nawet jeżeli nie wystąpiły błędy odwzorowania optycznego.

2.2. Charakterystyka obszarów obrazu zawierających tekst

Z reguły tekst musi spełniać kryterium czytelności. Intuicyjne cechy zapewniające czytelność tekstu, takie jak: kontrast barwy lub jasności w stosunku do tła, jednolitość barwy w obrębie tekstu jak i litery oraz ustawienie liter na jednej linii są stosowane podczas projektowania otaczających nas napisów [3]. Cechy te powinny zostać zachowane przez algorytm segmentacji, aby istniała możliwość ich wykorzystania podczas późniejszej analizy wyszukującej segmentów mogących być tekstem.

Algorytm segmentacji powinien być odporny na zakłócenia wynikające z niejednorodności oświetlenia napisu lub pojedynczej litery, co w warunkach naturalnego oświetlenia może zdarzać się często.

2.3. Wybór przestrzeni barwnej

Efektywna segmentacja obrazów z uwzględnieniem cech podanych w p. 2.1 i p. 2.2 wymaga zmiany sposobu kodowania koloru. Opisywany algorytm posługuje się przestrzenią Lab. Piksel w przestrzeni Lab opisywany jest za pomocą trzech wartości: L (jasność), a (równowaga pomiędzy barwą zieloną a czerwoną) i b (równowaga pomiędzy barwą niebieską a żółtą). Zaletą tej przestrzeni jest możliwość zdefiniowania odległości w przestrzeni barw zgodnej z intuicyjnym pojmowaniem barwy, oraz ciągłość: każde ciągłe przejście tonalne występujące w obrazie jest również ciągłe w przestrzeni Lab. Warunek ciągłości nie jest spełniony np. w przestrzeni HSL lub HSB.

2.4. Konwersja obrazu do przestrzeni Lab

Obraz uzyskiwany z aparatu cyfrowego najczęściej zapisany jest w formacie RGB. Konwersja do przestrzeni Lab jest dwuetapowa [1][2], i wymaga pośredniej konwersji do przestrzeni XYZ:

$$R,G,B \in \langle 0..1 \rangle$$

$$\text{if } R \le 0.0405 \text{ then } R:=\frac{R}{12.92} \text{ else } R:=\left(\frac{R+0.055}{1.055}\right)^{2.4}$$

$$\text{if } G \le 0.0405 \text{ then } G:=\frac{G}{12.92} \text{ else } G:=\left(\frac{G+0.055}{1.055}\right)^{2.4}$$

$$\text{if } B \le 0.0405 \text{ then } B:=\frac{G}{12.92} \text{ else } B:=\left(\frac{B+0.055}{1.055}\right)^{2.4} \qquad (2.4.1)$$

$$R:=R\times 100\,,\ G:=G\times 100,\ B:=B\times 100$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}:=\begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}\times\begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Konwersja do przestrzeni Lab odbywa się w następujący sposób:

$$x:=\frac{X}{95.047},\ y:=\frac{Y}{100},\ z:=\frac{Z}{108.883}$$

$$\text{if } x \le 0.008856 \text{ then } x:=x\times 7.787+\frac{16}{116} \text{ else } x:=\sqrt[3]{x}$$

$$\text{if } y \le 0.008856 \text{ then } y:=y\times 7.787+\frac{16}{116} \text{ else } y:=\sqrt[3]{y}$$

$$\text{if } z \le 0.008856 \text{ then } z:=z\times 7.787+\frac{16}{116} \text{ else } z:=\sqrt[3]{z} \qquad (2.4.2)$$

$$L:=116\times y-16$$

$$a:=500\times(x-y)$$

$$b:=200\times(y-z)$$

Powyższe przekształcenia są ważne dla obserwatora 2° i iluminantu D65.

2.5. Algorytm segmentacji

Pierwszym krokiem algorytmu segmentacji jest zastosowanie krawędziowego filtru Sobela [4] na kanale L obrazu źródłowego przekonwertowanego do systemu Lab. Filtr ten osobno wykrywa krawędzie pionowe oraz poziome poprzez sumowanie wartości pikseli otaczających piksel badany z wagami określonymi przez tablice [2]:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ oraz } \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (2.5.1)$$

Wyniki działania filtru dla krawędzi pionowych S_v i poziomych S_h wykorzystywane są do stworzenia obrazu krawędzi L^* zgodnie z wzorem:

$$L^*:=|S_v|+|S_h|-\left(\frac{|S_v|\times|S_h|}{128}\right) \qquad (2.5.2)$$

gdzie:

$|S_v|,|S_h|$:wartość bezwzględna wyjścia filtrów krawędziowych działających na składowej L obrazu źródłowego Lab dla $L \in \langle 0..100 \rangle$

Powyższa funkcja została wybrana na podstawie wizualnej oceny wyników.

W dalszym ciągu algorytm segmentacji wyszukiwać będzie spójnych kolorystycznie obszarów w kanałach ab oddzielonych krawędziami kanału L^*.

Właściwa segmentacja odbywa się kilku etapach: pierwszym jest znalezienie obszarów spełniających kryterium spójności. Dla dwóch pikseli kryterium spójności określone jest formułą:

$$K_s(p_1,p_2)=\begin{cases}1: \left\|ab_{p_1}-ab_{p_2}\right\| < f\left(L^*_{p_1}\right)\\ 0: \left\|ab_{p_1}-ab_{p_2}\right\| \geq f\left(L^*_{p_1}\right)\end{cases} \qquad (2.5.3)$$

gdzie:

$\left\|ab_{p_1}-ab_{p_2}\right\|$:euklidesowa odległość pikseli $p_1\, p_2$ na płaszczyźnie ab przestrzeni Lab

$L^*_{p_1}$: wartość na wyjściu filtru krawędziowego dla piksela p1

$f(\,)$:funkcja kryterium spójności

Zadaniem funkcji *f()* jest zmniejszenie wpływu szumów w obrazie krawędzi na działanie algorytmu.

Kryterium (2.5.3) stosowane jest w podanym poniżej algorytmie:

```
for y:=1 to Wysokość_Obrazu do
  for x:=1 to Szerokość_Obrazu do
   begin
    if K_s(p(x,y),p(x-1,y))
     then Tablica_Segmentów[x,y]:=Tablica_Segmentów[x-1,y]
     else begin
          if K_s(p(x,y),p(x,y-1))
           then Tablica_Segmentów[x,y]:=Tablica_Segmentów[x,y-1]
           else if K_s(p(x,y),p(x-1,y-1))
                then Tablica_Segmentów[x,y]:=Tablica_Segmentów[x-1,y-1]
                else
                 begin
                  Numer_Segmentu:=NumerSegmentu+1
                  Tablica_Segmentów[x,y]:=Numer_Segmentu
                 end
          end
   end
```

Algorytm 2.5.4

Wynikiem działania algorytmu (2.5.4) jest tablica segmentów będąca w istocie obrazem o rozmiarach identycznych z obrazem przetwarzanym, i wartościach pikseli odpowiadających numerom segmentów.

Segmenty zawarte w tej tabeli wymagają scalenia, co realizuje następujący algorytm:

```
repeat
  scalono_obszary:=false
  for x:=1 to Szerokość_Obrazu do
   for y:=1 to Wysokość_Obrazu do
    begin
      if {piksel(x,y) i piksel(x,y+1) należą do różnych segmentów C1 i C2} and
K_s(p(x,y),p(x,y+1)) then
       begin
        scalono_obszary:=true
        for x_1:=1 to Szerokość_Obrazu do
          for y_1:=1 to Wysokość_Obrazu do
           if Tablica_Segmentów[x_1,y_1]:=numer_segmentu_C2
                then Tablica_Segmentów[x_1,y_1]:=numer_segmentu_C1
       end
    end
until scalono_obszary=false
```

Algorytm 2.5.5

Otrzymany w wyniku segmentacji obraz posiada znaczną ilość segmentów składających się z małej ilości pikseli. Dlatego kolejnym krokiem segmentacji jest „przyklejenie" pikseli zawartych w małych segmentach do przyległych dużych segmentów. W przypadku graniczenia z kilkoma dużymi segmentami, wybierany jest wskazywany przez najbliższy w przestrzeni ab piksel będący bezpośrednim sąsiadem (b-sąsiadem)[5].

Ostatnim krokiem segmentacji jest uśrednienie kolorów wszystkich pikseli znajdujących się w poszczególnych segmentach. Składowe RGB średniego koloru segmentu obliczane są jako średnia arytmetyczna odpowiednich składowych R, G oraz B wszystkich pikseli należących do segmentu.

2.6. Przykładowy rezultat segmentacji

Opisany algorytm wykorzystany został do segmentacji obrazów pozyskanych z cyfrowego aparatu fotograficznego Casio QV4000 o rozdzielczości 4Mpx (2240x1680 pikseli). Zastosowana została funkcja kryterium spójności postaci

$$f(L^*) = 30 - \frac{3 \times L^{*2}}{255} \tag{2.6.1}$$

Obraz zapisany został w bezstratnym formacie TIFF z osadzonym profilem barwnym sRBG. Wynik zaprezentowany jest na fot.1.



fot.1 Oryginalny oraz poddany segmentacji obraz uzyskany z aparatu cyfrowego.

Obraz wynikowy charakteryzuje:
1 Brak szczegółów tła. Niemal całe tło jest scalane w pojedynczy segment, lub kilka segmentów o znacznej wielkości, co znacznie przyśpiesza późniejszą analizę kontekstową.
2 Brak efektów wynikających z lokalnych zmian jasności pikseli, spowodowane lokalnymi zmianami intensywności oświetlenia (napis „kantor" umieszczony na karniszu w lewej części fotografii). Pozwoli to na oszacowanie parametrów geometrycznych obiektów znajdujących się na obrazie analizowanym na podstawie wielkości segmentów.
3 Zachowanie kolorów oryginalnego obrazu w segmentach obrazu przetworzonego. Analiza kontekstowa segmentów może zatem uwzględniać kolory analizowanych elementów obrazu.

3. KONTEKSTOWA ANALIZA SEGMENTÓW OBRAZU WYNIKOWEGO

Analiza kontekstowa polega na porównaniu określonych parametrów segmentów obrazu.

3.1. Kryteria analizy kontekstowej

Obraz z wyodrębnionymi segmentami poddany został analizie mającej na celu poszukiwanie ciągów segmentów mogących stanowić tekst. Zastosowane zostały kryteria zgodne z intuicją:
1. litery stanowiące tekst muszą mieć podobną wysokość
2. wszystkie znaki w tekście powinny znajdować się w podobnej odległości od dolnej krawędzi obrazu
3. kolor wszystkich elementów napisu musi być zbliżony

Na potrzeby analizy kontekstowej przyjęto odpowiadające podanym kryteriom formuły wysokości segmentu:

$$c_1 > \frac{H_{s_k}}{H_{s_l}} > c_2 \qquad (3.1.1)$$

gdzie:

H_{s_k}, H_{s_l} : wysokość analizowanych segmentów w pikselach

c_1, c_2 : stałe, k,l :numer segmentu odległości od krawędzi obrazu:

$$\frac{\left|Y_{s_k} - Y_{s_l}\right|}{H_l} < c_3 \qquad (3.1.2)$$

gdzie:

Y_{s_k}, Y_{s_l} : odległość najniżej położonego piksela segmentu od dolnej krawędzi obrazu

H_l :wysokość obrazu w pikselach, c_3 :stała, k,l : numer segmentu

oraz bliskości kolorów:

$$\left|R_{s_k} - R_{s_l}\right| + \left|G_{s_k} - G_{s_l}\right| + \left|B_{s_k} - B_{s_l}\right| < c_4 \qquad (3.1.3)$$

gdzie:

R,G,B_{s_k}, R,G,B_{s_l} : składowe kolorystyczne RGB porównywanych segmentów

c_4 : stała

k,l : numery segmentów

Łączne kryterium K_k jest spełnione, gdy spełnione są kryteria (3.1.1), (3.1.2) i (3.1.3).

Zbiory segmentów tworzone są poprzez porównywanie zbioru zarodkowego $N_n := \{S_n\}$,gdzie S_n jest n-tym segmentem obrazu, z pozostałymi segmentami:

```
for n:=0 to max do
 begin
  N_n := {S_n}
  p := n+1
  repeat
   if ∃ K_k(S_z, S_p) then N_n := N_n ∪ {S_p}
   p:=p+1
   until p=max
  end

gdzie:
max : największy numer segmentu
N_n : zbiór segmentów mogących być tekstem
S_n : segment o numerze n
```

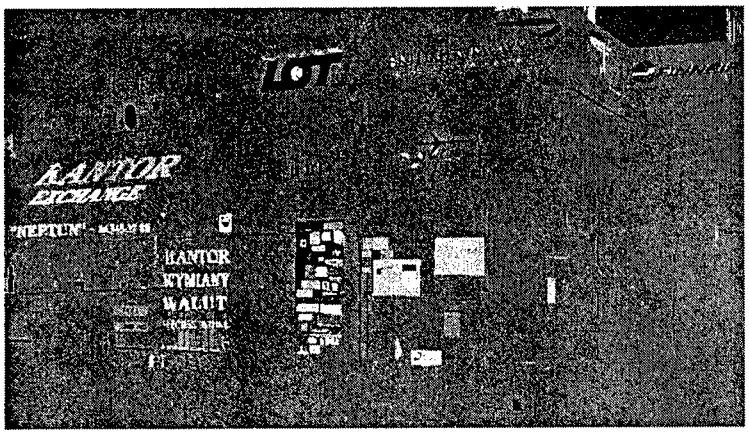
Algorytm (3.1.4)

W celu przyśpieszenia pracy algorytmu zostało wprowadzone ograniczenie na wielkość analizowanych segmentów.

3.2. Przykładowy wynik analizy kontekstowej

Przykładowe wyniki fotografii z fot. 1 zaprezentowane są na fot. 2.

Dla pokazanego na fot.1 obrazu wykonana została segmentacja według opisanego algorytmu, a następnie analiza kontekstowa opisana w p. 3.1. Przyjęto parametry c_1=0.85, c_2=1.1765, c_3=0,02 oraz c_4=35. Algorytm znajdował zbiory segmentów spełniających kryterium kontekstowe K_k.



fot.2 Wynik analizy kontekstowej obrazu z fot.1

Następnie dla każdego segmentu należącego do zbioru wyznaczony został punkt charakterystyczny o rzędnej równej najmniejszej wartości rzędnej wszystkich pikseli segmentu, i odciętej równej najmniejszej wartości odciętej wszystkich pikseli segmentu.

Poprowadzenie dla każdego zbioru odcinka pomiędzy skrajnymi – lewym i prawym – punktami charakterystycznymi spowodowało podkreślenie segmentów mogących stanowić napis. Zaznaczone zostały jedynie zbiory 3 i więcej elementowe.

Dobór współczynników $c_{1..4}$ ma istotny wpływ na jakość pracy algorytmu.

4. ZAKOŃCZENIE

Zaprezentowana metoda poszukiwania informacji tekstowej w obrazie stanowi punkt wyjściowy dla dalszych badań.

Obraz poddany segmentacji wymaga dodatkowej obróbki, pozwalającej na sprawniejsze wykrywanie segmentów powstających z wyodrębnienia z obrazu liter pochodzących z „lekkich" zbiorów czcionek. Jest prawdopodobne, że sposób tej obróbki zależny będzie w istotny sposób od zastosowanego źródła obrazów.

Zaprezentowana metoda analizy kontekstowej tworzy zbiory segmentów w oparciu o kilka prostych kryteriów. Sprecyzowanie kryteriów tworzenia zbiorów segmentów pozwoli na lepsze rozpoznawanie obszarów zawierających tekst. Ponadto analiza parametrów elementów zbiorów pozwoli na ustalenie charakteru zmian geometrycznych liter wynikających z powstałych w wyniku odwzorowania optycznego skrótów perspektywicznych.

BIBLIOGRAFIA

[1] Adobe Photoshop 4.0 Software Development Kit, Adobe Systems Incorporated, 30 October 1996.
[2] Pitas I.: Digital image processing algorithms. Prentice Hall, 1993
[3] Wheeler S. G., Wheeler G.S.: Typografia komputerowa. Exit, Warszawa
[4] Jain R., Kasturi R.,Schnuck B.: Machine vision. McGRAW-HILL, 1995
[5] Pavlidis T., *Grafika i przetwarzanie obrazów*. WNT, Warszawa 1987

autorem fotografii jest Piotr Zabłocki

COLOR-PRESERVING IMAGE SEGMENTATION AND ITS APPLICATION TO LOCALIZATION OF REGIONS CONTAINING TEXT

Summary

Text contained in images we percept in our environment is very important source of information. It's obvious that blind persons cannot use them. On the other hand, there is a number of OCR programs and voice synthesizers. Unfortunately, OCR-s are not able to recognize text delivered by cameras from surrounding world properly. But image segmentation algorithm in connection with context image segmentation is able to detect text containing regions of image and prepare them for OCR.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Roman Rykaczewski

Katedra Systemów Informacyjnych, Politechnika Gdańska

DWUPOZIOMOWE CYFROWE ZNAKOWANIE OBRAZÓW NIERUCHOMYCH

Streszczenie

W pracy przedstawiono metodę cyfrowego znakowania obrazów nieruchomych na dwóch poziomach. Cyfrowy znak wodny wprowadzony na pierwszym poziomie jest znakiem wrażliwym, pozwalającym na szybkie wykrycie manipulacji na obrazie. Znakowanie na drugim poziomie jest znakowaniem silnym (odpornym) w tym sensie, że jego usunięcie związane jest z tak dużym zniekształceniem obrazu, że staje się on bezużytecznym. Znakowanie to jest zabezpieczone kodem korekcyjnym Reeda-Solomona. W procesie znakowania odpornego i projektowania kodu zabezpieczającego wykorzystywane są właściwości statystyczne obrazów na najbardziej znaczących płaszczyznach bitowych.

1. WSTĘP

Niezwykle szybki w ostatnich dwóch dziesięcioleciach rozwój techniki i technologii cyfrowych spowodował, że praktycznie każdy ma nie tylko dostęp do różnego rodzaju danych zapisanych cyfrowo, takich jak dźwięki, obrazy nieruchome, obrazy ruchome, teksty, oprogramowanie itp., ale także ma niezwykle duże możliwości ich przetwarzania i modyfikowania. Stwarza to szereg nie występujących wcześniej tak ostro zagrożeń związanych z zabezpieczeniem integralności informacji, praw autorskich oraz uwierzytelniania źródła pochodzenia informacji. Środkiem zabezpieczającym przed tymi zagrożeniami jest wprowadzanie do zabezpieczanych informacji tzw. cyfrowego znaku wodnego, mającego postać, najczęściej niewidzialnego, w przypadku obrazów, lub nie dającego się usłyszeć, w przypadku dźwięków, specjalnie uformowanego sygnału cyfrowego. Sygnał ten, z punktu widzenia odbiorcy informacji, stanowi oczywiście zakłócenie i w związku z tym jego poziom nie może być zbyt duży. Z drugiej strony, aby znak wodny mógł spełniać swoje zadanie, sygnał ten musi być odporny na ataki osób chcących go usunąć, a to z kolei jest tym trudniejsze im poziom sygnału znaczącego jest wyższy. Podobnie jak przy budowaniu zabezpieczeń kryptograficznych, przy projektowaniu zabezpieczenia znakiem wodnym przyjmuje się założenie, że atakujący zna metodę znakowania, a nie jest mu znany jedynie tajny klucz pozwalający na odczytanie treści znaku, czy też tylko na stwierdzenie jego obecności w analizowanych danych. W literaturze opisano wiele metod znakowania i ciągle powstają nowe rozwiązania [1],[2]. Równocześnie powstają też nowe sposoby

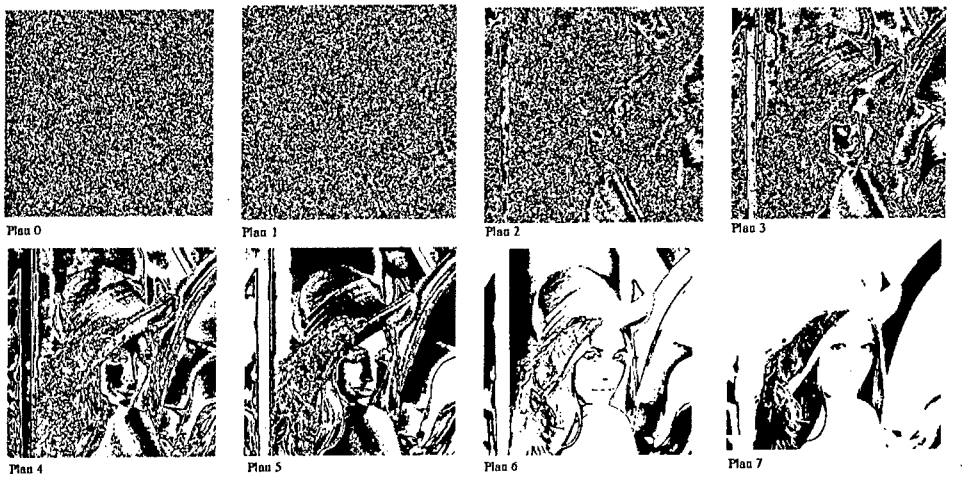
ataku na zabezpieczenia, często bardzo proste w realizacji i jednocześnie skuteczne – tzn. usuwają one znak lub powodują, że nie jest możliwe jego odczytanie, czy też stwierdzenie obecności bez znaczącego, zauważalnego naruszenia zabezpieczanej informacji [3],[4]. Można z dużą pewnością stwierdzić, że w chwili obecnej nie istnieje jedna metoda, która zabezpieczałaby przed wszystkimi, znanymi metodami ataku. Istniejące metody znakowania zabezpieczają jedynie przed niektórymi, wybranymi atakami.

W tej pracy zaproponowana jest nowa, prosta metoda dwustopniowego zabezpieczania obrazów nieruchomych. Dla uproszczenia opisu i ze względu na ograniczoną objętość artykułu działanie metody zostało przedstawione na przykładzie obrazu monochromatycznego, ale metoda ta może być zastosowana także do obrazów kolorowych. Pierwszym stopniem zabezpieczenia jest znak wrażliwy (ang. *fragile*). Celem umieszczenia tego znaku jest zapewnienie możliwości szybkiego, mało skomplikowanego wykrycia manipulacji na obrazie. Drugim stopniem zabezpieczenia jest znak odporny (ang. *robust*), którego zadaniem jest zapewnienie możliwości udowodnienia źródła pochodzenia obrazu nawet przy bardzo silnym, prowadzącym do widocznych zmian w wyglądzie obrazu, ataku na zabezpieczenie. Proponowana metoda znakowania należy do znakowania w dziedzinie przestrzeni - do znakowania wykorzystano plany bitowe obrazu wybrane na podstawie badania wrażliwości bitów poszczególnych planów na kompresję stratną JPEG.

W rozdziale 2. opisano metodę znakowania wrażliwego, polegającego na umieszczeniu prostego czarno-białego obrazu na jednym lub kilku mniej znaczących planach bitowych. Następnie, w rozdziale 3. zaproponowano metodę znakowania odpornego, zabezpieczonego kodem korekcyjnym Reeda-Solomona i przeprowadzono analizę parametrów takiego kodu.

2. ZNAKOWANIE WRAŻLIWE

Najprostszym sposobem umieszczenia sygnału znaku wrażliwego jest zapisanie go na najmniej znaczących bitach liczb zawierających wartości pikseli obrazu. Na rys.1 pokazano obrazy ośmiu planów bitowych obrazu testowego Lena.

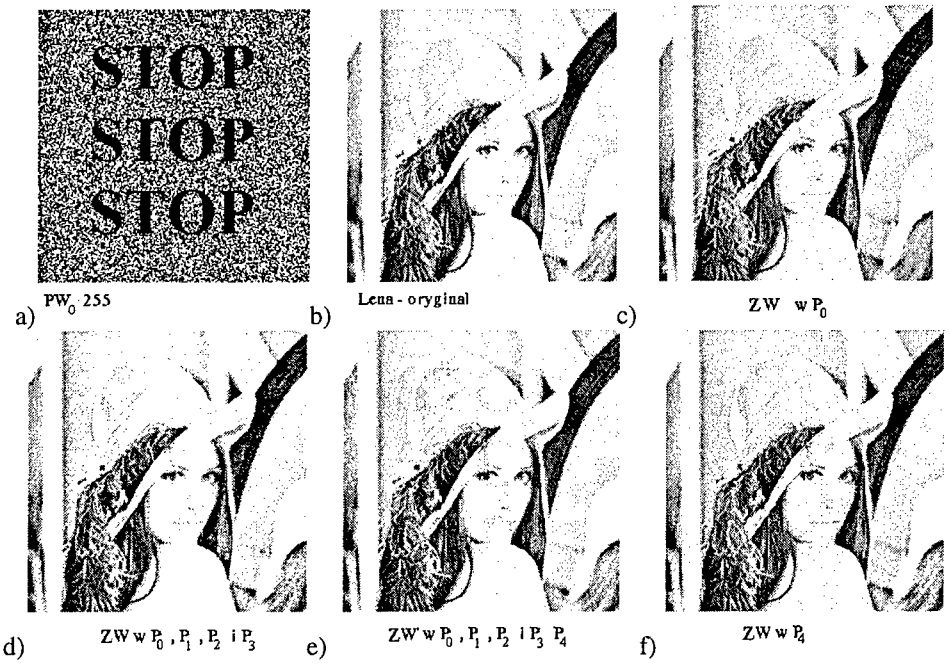


Rys.2.1. Plany bitowe obrazu testowego Lena

Jak widać z rysunku, losowy obraz zawartości planów 0. i 1 sugeruje, że umieszczenie sygnału znaku na tych planach nie powinno w widoczny sposób wpłynąć na wygląd obrazu. Również na planach 2. i 3. istnieją duże obszary, w których umieszczenie znakowania nie wpłynie na jakość wizualną obrazu. Plany bitów na bardziej znaczących pozycjach charakteryzują się występowaniem w nich wyraźnych krawędzi obiektów będących ważnym elementem treści obrazu i niezauważalne umieszczenie tam informacji znakującej jest utrudnione. W związku z powyższą naturą planów bitowych, która jest właściwa dla klasy analizowanych w tej pracy obrazów (obrazy o 256 odcieniach szarości i o „urozmaiconej" strukturze), naturalnym jest umieszczenie znaku wrażliwego na planie zerowym (czyli bitów najmniej znaczących) i ewentualnie na planach do 3. włącznie, co pozwala na uzyskanie mocniejszego, bardziej odpornego na zniekształcenia znakowania. Na rys.2.2 pokazano obraz testowy oznaczony znakiem wrażliwym (ZW), mającym postać obrazu zawierającego czarne napisy **STOP** na białym tle, na planach od 0. do 3.(na rys.2.2e,f także planu 4. dla pokazania, że prowadzi to do uwidocznienia obrazu znaku). Znak wrażliwy został umieszczony na planach bitowych zgodnie z wzorem

$$\mathbf{PW}_i = \mathbf{P}_i \vec{\otimes} \mathbf{W}_i \; ; \quad i = 0,1,2,3 \tag{2.1}$$

gdzie $\vec{\otimes}$ oznacza operację po-elementowego iloczynu (ang. *component-wise product, dot product, Schur product*) modulo 2 macierzy $\mathbf{P}_i$ planu bitowego i oraz macierzy $\mathbf{W}_i$ będącej obrazem znaku wrażliwego dla planu i (w ogólnym przypadku znaki dla poszczególnych planów mogą być różne).



Rys.2.2. Obraz testowy Lena oznakowany znakiem wrażliwym: a) plan P_0 (LSB) z wprowadzonym znakiem wrażliwym, pomnożony przez 255, b) oryginalny obraz testowy, c) obraz testowy oznakowany na planie P_0, d) obraz testowy oznakowany tym samym znakiem na planach P_0, P_1, P_2 i P_3, e) obraz testowy oznakowany tym samym znakiem na planach od P_0 do P_4, f) obraz testowy oznakowany tylko na planie P_4.

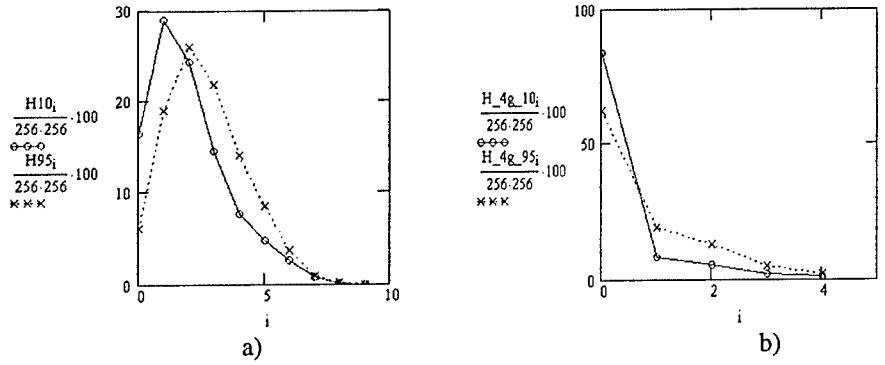
Odczytanie znaku wrażliwego, przy założeniu, że nie dysponujemy obrazem oryginalnym, sprowadza się do wydzielenia z badanego obrazu jednego lub większej liczby planów bitowych i zobrazowanie każdego planu przez pomnożenie go przez 255. Przy braku manipulacji na obrazie, obraz znaku wodnego będzie wyraźnie widoczny. Zastosowanie do oznaczonego obrazu nawet nieznacznego stopnia kompresji powoduje zatarcie oznakowania. Jeżeli dysponujemy obrazem oryginalnym, to wykrycie znaku wrażliwego, jednakowego dla wszystkich wykorzystywanych planów jest jeszcze łatwiejsze- wystarczy odjąć od obrazu oryginalnego obraz oznakowany i macierz wynikową pomnożyć przez 255.

Przy zastosowaniu jednakowych obrazów znakujących na planach od 0. do 3. otrzymujemy możliwość wykorzystania operacji ich uśrednienia do ewentualnego wydobycia oznakowania po zastosowaniu kompresji stratnej na małym poziomie.

3. ZNAKOWANIE ODPORNE

Znak odporny musi być umieszczony w tych planach bitowych, które zawierają najistotniejszą oglądowo informację, czyli planach od 4. do 7. Sekwencję znaków alfanumerycznych, która zawiera istotne dane obrazu i jego autora zapisujemy w postaci ciągu binarnego. Ciąg ten, stanowiący znak odporny, dzielimy na podciągi 4. bitowe (pół-bajty) i poszukujemy w zbiorze bardziej znaczących pół-bajtów wartości pikseli obrazu pół-bajtów identycznych lub możliwie jak najmniej różniących się od pół-bajtów znaku i zastępujemy je pół-bajtami znaku odpornego, zapisując jednocześnie ich pozycje w obrazie. Informacja o pozycjach, w których nastąpiła podmiana pół-bajtów obrazu na pół-bajty znaku stanowi, łącznie z ciągiem znaku, tajny klucz metody.

Zastosowanie kompresji stratnej jako metody ataku na oznakowanie może spowodować, że także w bardziej znaczących pół-bajtach pojawią się błędne bity. Na rys.3.1 pokazano wykresy wagowego widma błędów (rozkład częstości względnych liczby błędów w ciągach binarnych o ustalonej długości, charakterystyka powszechnie używana w opisie właściwości statystycznych kanałów cyfrowych) dla bitów w planach od 0. do 7. – rys.3.1.a) – pokazujące procent liczby bajtów zawierających od 0. do 8. błędnych bitów, po dokonaniu kompresji stratnej na poziomie 10% i 95% oraz na rys.3.1.b) podobne wykresy dla bardziej znaczących pół-bajtów.



a) b)

Rys.3.1. Wpływ stopnia kompresji stratnej JPEG (na poziomie 10% oraz 95%) obrazu testowego Lena na liczbę błędnych bitów: a)-procent liczby bajtów zawierających od 0 do 8. błędnych bitów; b)- procent liczby bardziej znaczących pół-bajtów zawierających od 0 do 4. błędnych bitów.

Widoczne jest, że stopień kompresji stratnej JPEG ma stosunkowo niewielki wpływ na rozkład liczby błędów w bardziej znaczących pół-bajtach – w tabeli 3.1 podano rozkład liczby bardziej znaczących pół-bajtów zawierających 0,1,2,3,4 błędy dla kompresji JPEG obrazu testowego Lena dla stopni kompresji 10% i 95%. Aby zabezpieczyć się przed błędami, jakie w znaku wodnym może wprowadzić atak na oznakowanie za pomocą kompresji stratnej JPEG stosujemy kodowanie nadmiarowe informacji zawartej w znaku odpornym. W tym celu ciąg bajtów znaku odpornego traktujemy jako ciąg bajtów informacyjnych i dokonujemy kodowania blokowym kodem Reeda-Solomona, dopisując ciąg bajtów kontrolnych.

Tabela 3.1

Rozkład liczby bardziej znaczących pół-bajtów zawierających 0,1,2,3,4 błędy dla kompresji JPEG obrazu Lena stopnia 10% i 95%.

Stopień kompresji	10%				
Liczba błędów	0	1	2	3	4
Procent liczby pół-bajtów	83,7	8,0	5,4	1,9	1,0
Stopień kompresji	95%				
Procent liczby pół-bajtów	61,9	18,9	12,6	4,7	1,9

Bajty kontrolne można umieścić w bardziej znaczących pół-bajtach znakowanego obrazu na podobnej zasadzie jak bajty informacyjne znaku i zapisać ich pozycje w kluczu. Przy dostatecznie dużych rozmiarach znaczonego obrazu możliwe jest znalezienie wielu pół-bajtów identycznych z pół-bajtami znaku lub mało różniących się wartością. Ich uwzględnienie w kluczu dałoby dodatkowe zabezpieczenie przed atakami geometrycznymi, polegającymi na usuwaniu, czy przestawianiu pojedynczych linii w obrazie.

Okazuje się jednak, że procent liczby bezbłędnych bardziej znaczących pół-bajtów, nawet przy stosunkowo niewielkim stopniu kompresji 10% jest zbyt mały, aby zastosowany kod nadmiarowy, przy niezbyt dużym nadmiarze kodowym, miał możliwość korekcji najbardziej prawdopodobnej liczby błędów. W związku z tym do zapisania znaku odpornego należy wykorzystać, mniej wrażliwe na ataki, dwa najbardziej znaczące bity, planu 6. i planu 7.

W tabeli 3.2 pokazano rozkład liczby 2. najbardziej znaczących bitów (dibitów) zawierających 0,1,2 błędy dla kompresji JPEG obrazu Lena stopnia 10% i 95%.

Tabela 3.2

Rozkład liczby 2. najbardziej znaczących bitów zawierających 0,1,2 błędy dla kompresji JPEG obrazu Lena stopnia 10% i 95%.

Stopień kompresji	10%		
Liczba błędów	0	1	2
Procent liczby pół-bajtów	97,1	1,9	1,0
Stopień kompresji	95%		
Procent liczby pół-bajtów	92,6	4,6	2,8

3.1. Wybór parametrów kodu Reeda-Solomona

Podane wyżej, przykładowe wyniki badania wrażliwości najbardziej znaczących bitów wartości pikseli obrazu na kompresję JPEG oraz obliczone rozkłady liczby bardziej znaczących pół-bajtów oraz liczby par najbardziej znaczących bitów z błędami powstałymi wskutek zastosowania kompresji stratnej JPEG pozwalają na oszacowanie parametrów kodu Reeda-Solomona (R-S) zabezpieczającego znak odporny.

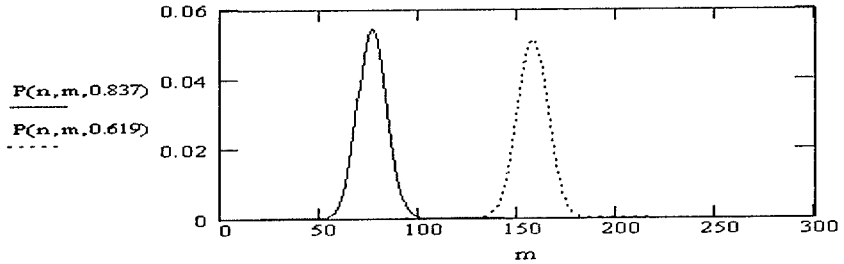
Kody R-S są podklasą niebinarnych kodów liniowych, których symbolami są elementy ciała Galois CG(q), gdzie q jest całkowitoliczbową potęgą liczby pierwszej p. Kod R-S (n,k), o długości ciągów kodowych n, liczbie pozycji informacyjnych k i korygujący t błędnych symboli z CG(q) ma następujące parametry [5]:

- długość ciągu kodowego: $n = q - 1$;
- liczba elementów kontrolnych: $n - k = 2\,t$;
- minimalna odległość kodowa: $d_{min} = 2\,t + 1$.

Przyjmiemy tu, że $p = 2$ i $q = p^8$=256 (elementami ciągu kodowego R-S są bajty) i w związku z tym długość ciągów kodowych wynosi n = 255. Najpierw zajmiemy się opisanym w p.2.1 sposobem wykorzystania bardziej znaczących pół-bajtów. W tym przypadku każdy element ciągu kodowego kodu R-S, stanowiący znak odporny, składałby się z dwóch, bardziej znaczących pół-bajtów wartości pikseli obrazu. Jeżeli prawdopodobieństwo wystąpienia bezbłędnego pół-bajtu, przy maksymalnym, dopuszczalnym ze względu na występowanie widocznych zniekształceń, stopniu kompresji, oznaczymy $P_{4\text{-}0}$, to prawdopodobieństwo, że pojedynczy bajt w ciągu kodowym będzie bezbłędny jest równe $P_{4\text{-}0}^2$. Będzie tak, przy założeniu, że zdarzenia polegające na porażeniu błędami pół-bajtów są niezależne statystycznie. Ponieważ pół-bajty składające się na jeden bajt-element ciągu kodowego- będą pochodziły, na ogół, z różnych miejsc płaszczyzny obrazu, zatem założenie takie jest usprawiedliwione. Bazując na tym założeniu możemy w prosty sposób, korzystając z rozkładu dwumianowego, obliczyć prawdopodobieństwo $P_n(m)$, że w ciągu kodowym, złożonym z n bajtów wystąpi m bajtów z błędami (wagowe widmo błędnych bajtów w ciągu kodowym kodu R-S):

$$P_n(m) = \binom{n}{m} \left(1 - P_{4\text{-}0}^{\,2}\right)^m \left(P_{4\text{-}0}^{\,2}\right)^{n-m} \qquad (3.1)$$

Na rys. 3.4. pokazano wykresy prawdopodobieństwa $P_n(m)$ (oznaczonego na rysunku jako P(n,m,$P_{4\text{-}0}$)) dla obrazu testowego Lena i dwóch stopni kompresji: 10% ($P_{4\text{-}0}$=0,837) i 95% ($P_{4\text{-}0}$=0,619).



Rys.3.4. Wykresy prawdopodobieństwa $P_n(m)$ przy wykorzystaniu pół-bajtów dla obrazu testowego Lena i dwóch stopni kompresji: 10%- linia ciągła i 95% – linia przerywana.

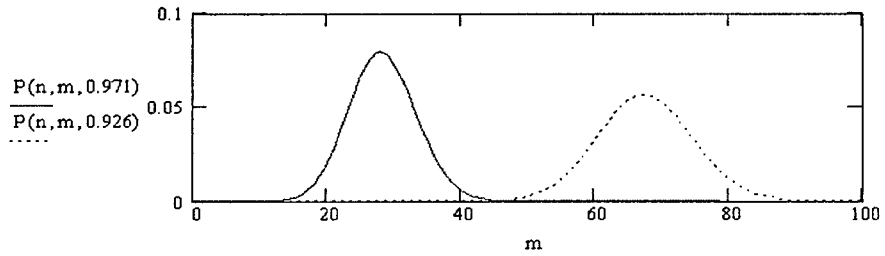
Widoczne jest, że już przy małym stopniu kompresji – 10%, praktycznie pewnym jest wystąpienie od 55 do 100 błędnych bajtów w ciągu kodowym o długości 255 bajtów. Oznacza to, że dla ich korekcji konieczne byłoby zastosowanie kodu z liczbą pozycji kontrolnych równą: $n - k = 2 \cdot t = 2 \cdot 100 = 200$. W związku z tym liczba pozycji informacyjnych (długość znaku odpornego) byłaby równa: $k = n - 200 = 55$. Dla wyższych stopni kompresji długość znaku odpornego byłaby jeszcze mniejsza. Zwiększenie długości ciągu informacyjnego można uzyskać poprzez zwiększenie liczby q np. do wartości

$q=2^{12}$ =4096. Długość ciągu kodowego byłaby wówczas równa n=4095 i pojedynczy symbol kodowy składałby się z półtora bajtu. Jednak, ponieważ we wzorze (3.1) prawdopodobieństwo bezbłędnego bajtu byłoby teraz równe $P_{4\text{-}0}{}^3$, zatem można obliczyć, że dla stopnia kompresji 10% długość ciągu informacyjnego byłaby równa $k = n$–3600=495 bajtów. Należy jednak pamiętać, że znacznemu zwiększeniu złożoności uległaby procedura dekodowania tak długiego kodu.

Prostszym rozwiązaniem jest wykorzystanie małej wrażliwości dwóch najbardziej znaczących bitów każdego bajtu wartości pikseli obrazu. Przy ich wykorzystaniu, każdy symbol kodowy (bajt) składałby się z czterech par najbardziej znaczących bitów każdego bajtu wartości wybranych pikseli obrazu. W związku z tym prawdopodobieństwo $P_n(m)$, że w ciągu kodowym, złożonym z n bajtów wystąpi m bajtów z błędami byłoby teraz równe:

$$P_n(m)=\binom{n}{m}\left(1-P_{2\text{-}0}{}^4\right)^m\left(P_{2\text{-}0}{}^4\right)^{n-m} \qquad (3.2)$$

gdzie $P_{2\text{-}0}$ oznacza prawdopodobieństwo wystąpienia bezbłędnej pary najbardziej znaczących bitów wartości pikseli obrazu po zastosowaniu kompresji JPEG. Na rys. 3.5. pokazano wykresy prawdopodobieństwa $P_n(m)$ (oznaczonego na rysunku jako $P(n,m,P_{2\text{-}0})$), w takiej sytuacji, dla obrazu testowego Lena i dwóch stopni kompresji: 10% ($P_{2\text{-}0}$=0,971) i 95% ($P_{4\text{-}0}$=0,926).



Rys.3.5. Wykresy prawdopodobieństwa $P_n(m)$ przy wykorzystaniu dwóch najbardziej znaczących bitów dla obrazu testowego Lena i dwóch stopni kompresji: 10% – linia ciągła i 95% – linia przerywana.

Jak widać z rysunku, przy stopniu kompresji 10%, kod R-S z ciągiem kontrolnym o długości 2· 45=90, a więc przy długości znaku odpornego równej k=255-90=165 bajtów, zapewniłby korekcję prawie wszystkich, uszkodzonych wskutek kompresji, bajtów. Przy założeniu nawet tak silnej kompresji, jak kompresja 95%, ciąg kontrolny musiałby mieć długość 180 bajtów, a więc znak odporny musiałby mieć długość 75 bajtów. Ponieważ w wielu zastosowaniach 75 bajtów powinno wystarczyć na zapisanie danych identyfikacyjnych obrazu, zatem zastosowanie do zabezpieczenia znaku odpornego kodu R-S (255,75) i wykorzystania do konstrukcji elementów ciągu kodowego najbardziej znaczących dibitów bajtów wartości pikseli znaczonego obrazu będzie z nadmiarem przeciwdziałać atakom z wykorzystaniem kompresji stratnej JPEG.

4. ZAKOŃCZENIE

W pracy zaproponowano prostą implementacyjnie metodę zabezpieczania obrazów cyfrowym znakiem wodnym. Zabezpieczenie jest dwustopniowe. Znak wrażliwy, mający postać obrazu czarno-białego jest wprowadzany do najmniej znaczących planów bitowych znakowanego obrazu. Znak odporny, mający postać ciągu bajtów, jest projektowany dla uzyskania zabezpieczenia przed atakiem za pomocą kompresji stratnej JPEG i jest zabezpieczony za pomocą kodu korekcyjnego Reeda-Solomona. Do ujawnienia znaku odpornego konieczna jest znajomość tajnego klucza, który zawiera zapis pozycji dibitów tworzących ciąg kodowy kodu R-S oraz dane umożliwiające przeprowadzenie dekodowania kodu R-S. Stwierdzenie obecności znaku wrażliwego wymaga jedynie wydzielenia odpowiedniego planu bitowego z analizowanego obrazu lub, przy zapisie tego samego znaku wrażliwego na kilku planach, obliczenie wartości średniej tych planów. Zaproponowana metoda znakowania może być w prosty sposób rozszerzona na obrazy kolorowe, przez jej zastosowanie do macierzy wartości pikseli trzech składowych barwnych R, G, B. Jest to przedmiotem aktualnie prowadzonych badań.

Autor dziękuje anonimowemu recenzentowi za niezwykle wnikliwą opinię i cenne uwagi.

BIBLIOGRAFIA

[1] Min Wu, Bede Liu: *Data Hiding in Image and Video: Part I-Fundamental Issues and Solutions*, IEEE Transactions on Image Processing, vol. 12 (2003), nr 6, s.685-695.

[2] Min Wu, Bede Liu: *Data Hiding in Image and Video: Part II-Designs and Applications*, IEEE Transactions on Image Processing, vol. 12 (2003), nr 6, s.696-704.

[3] Voloshynovskiy S., Pereira S., Pun T., Eggers J., Su J. K.: *Attacks on Digital Watermarks: Classification, Estimation-based Attacks, and Benchmarks*, IEEE Communications Magazine, August 2001, s.118-125.

[4] Kutter M., Petitcolas F.A.P.: *A fair benchmark for image watermarking systems*, http://citeseer.ist.psu.edu/kutter99fair.html

[5] Lin S., Costello D. J.: *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1983.

TWO-LEVEL DIGITAL WATERMARKING OF STILL IMAGES

Summary

In this paper an image watermarking method for still images is proposed and described. The method uses two kinds of watermarks in image space domain. The first fragile watermark is embedded into least significant bit plane of the image or simultaneously into all three least significant planes of the image. The second watermark is robust to lossy JPEG compression. It consists of a chain of bytes, each built of four most significant dibits of the image selected pixel bytes. The strong watermark is protected by R-S error correcting code designed on the base of statistical analysis of the image most significant bit planes sensitivity to lossy JPEG compression. The secret key of watermarking method consists of dibit positions and R-S code generator polynomial coefficients.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Wojciech Seliga, Mariusz Szwoch

Katedra Inżynierii Wiedzy, Wydział ETI, Politechnika Gdańska

EKSTRAKCJA ORAZ DOPASOWYWANIE KONTURÓW MELODYCZNYCH W BAZACH DANYCH MUZYCZNYCH

Streszczenie

W pracy przedstawiono wybrane algorytmy indeksowania i wyszukiwania danych muzycznych w multimedialnych bazach danych. Opisano różnorodne techniki wprowadzania zapytań muzycznych, podając odpowiednie dla nich metody ekstrakcji melodii. Jako mechanizm indeksowania melodii zaproponowano kontury muzyczne. Zaimplementowana w ramach eksperymentów przeglądarka *MelodyExplorer* pozwala na praktyczną weryfikację przyjętych założeń oraz na przeprowadzenie eksperymentów w zakresie wprowadzania danych muzycznych z wielu źródeł, ekstrakcji melodii oraz porównywania konturów muzycznych. Przeglądarka współpracuje z licznymi urządzeniami zewnętrznymi komputera oraz programem *ScoreExplorer* rozpoznającym drukowaną notację muzyczną. Na zakończenie, przedstawiono wstępne wyniki wyszukiwania danych muzycznych przy użyciu systemu.

1. WSTĘP

Jedną z istotnych dziedzin zastosowań algorytmów przetwarzania i rozpoznawania danych są multimedialne bazy danych. Algorytmy tam stosowane służą nie tylko do pozyskiwania i opisu danych, ale są również wykorzystywane do tworzenie zapytań multimedialnych w celu wydobycia zgromadzonych danych. Niebagatelną rolę odgrywa też stworzenie właściwego interfejsu użytkownika, pozwalającego na stosowanie różnorodnych technik tworzenia zapytań multimedialnych zależnych od konfiguracji sprzętowej oraz umiejętności osoby tworzącej zapytanie. Kolejnym wreszcie aspektem jest problem wyszukiwania w multimedialnej bazie danych, a więc indeksowania danych oraz dopasowania zapytań do indeksów. W praktyce metody wyszukiwania muszą uwzględniać zarówno niedoskonałość algorytmów rozpoznawania i dopasowywania, jak również błędy powstające przy samym tworzeniu zapytań przez użytkownika.

Wyszukiwanie zorientowane na zawartość (*content-based retrieval*) jest obecnie przedmiotem wielu badań naukowych z uwagi na potencjalnie szerokie dziedziny zastosowań. W przypadku danych muzycznych obejmują one przykładowo: identyfikację utworów na podstawie fragmentów muzyki, badanie podobieństwa utworów, analizę muzykologiczną utworów czy wręcz szukanie plagiatów.

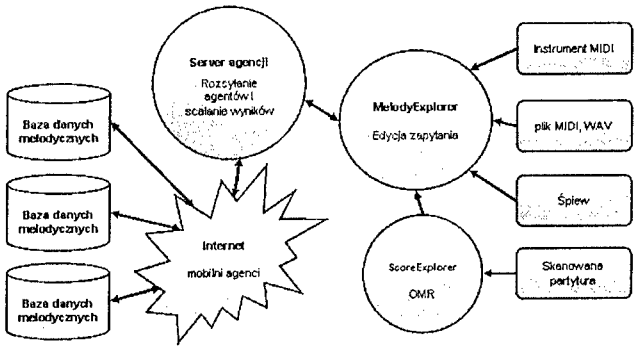
W niniejszej pracy przedstawiono propozycję organizacji multimedialnej bazy danych muzycznych oraz multimedialnego interfejsu tworzenia zapytań muzycznych. Szczególny nacisk położono na różnorodność metod wprowadzania danych oraz generowania zapytań. Metodami pozyskiwania danych mogą być przykładowo: systemy rozpoznawania notacji muzycznej (OMR – *Optical Music Recognition*), rozpoznawanie śpiewu, nucenia oraz gwizdania, komunikacja z urządzeniami MIDI, czy wreszcie edycja notacji muzycznej. Jako podstawową cechę indeksowania (i porównywania) wybrano kontury (profile) muzyczne melodii. W pracy przedstawiona została również programowa implementacja powyższych idei w postaci systemu *MelodyExplorer* oraz obiecujące wyniki pierwszych eksperymentów.

2. PROJEKT MUZYCZNEJ BAZY DANYCH

Bazą danych multimedialnych nazywamy komputerowy system baz danych, który umożliwia przechowywanie oraz wyszukiwanie według różnych kryteriów informacji multimedialnej, a więc nie tylko tekstu, ale również obrazów, dźwięków, czy filmów. Gwałtowny rozwój sprzętu komputerowego, technologii wytwarzania oprogramowania, algorytmów przetwarzania i rozpoznawania danych multimedialnych oraz Internetu umożliwił w pełni funkcjonalną realizację idei tworzenia bibliotek cyfrowych [1]. W ostatnich latach powstało wiele tego typu przedsięwzięć takich, jak np. Biblioteka Kongresu USA, Biblioteka Watykańska, Biblioteka Tajwańska oraz komercyjne biblioteki Virage i Excalibur.

Niestety, wśród bibliotek cyfrowych niezmiernie rzadko znajdują się biblioteki muzyczne, zawierające nie tylko zeskanowane obrazy partytur, ale również nagrania dźwiękowe. Jedyną powszechnie znaną biblioteką cyfrową tego typu jest nowozelandzki MELDEX, zawierający ok. 10 000 piosenek ludowych zapisanych w postaci plików MIDI [2,3]. System MELDEX umożliwia wyszukiwanie danych na podstawie przesłanego pliku dźwiękowego, zawierającego melodię zanuconą przez użytkownika. Niestety, ten sposób podawania zapytań mocno ogranicza funkcjonalność systemu. Jego kolejnym ograniczeniem jest rozpoznawanie i porównywanie wyłącznie zapisów monofonicznych.

Przedstawione w niniejszej pracy propozycje rozwiązań, potwierdzone wynikami wstępnych eksperymentów, pozwalają na zniesienie ograniczeń odnośnie formatu przechowywanych danych muzycznych (melodycznych), jak również ograniczeń w interfejsie wprowadzania zapytań do bazy danych.



Rys. 1. Formułowanie zapytania i wyszukiwanie w muzycznej bazie danych

Ogólny schemat systemu wyszukiwania informacji muzycznej przedstawiono na Rys. 1. Na podstawie wprowadzonych danych system dokonuje ich konwersji na postać monofoniczną MIDI i prezentuje je w interaktywnym edytorze notacji muzycznej. Po ich ewentualnej modyfikacji (edycji) i zatwierdzeniu przez użytkownika, następuje wyznaczenie zapytania w postaci konturu wprowadzonej melodii (por. 4).

W tradycyjnej konfiguracji, moduł zapytań jest bezpośrednio połączony z jedną multimedialną bazą danych oferując wyszukiwanie danych w trybie *on-line*. Autorzy proponują alternatywną konfigurację systemu, w której moduł zapytań komunikuje się z tzw. *agencją wyszukiwania*, korzystającą z technologii mobilnych agentów [4,5]. Konfiguracja taka daje możliwość współpracy z wieloma multimedialnymi bazami danych muzycznych, a jej dodatkową zaletą jest możliwość pracy w trybie *off-line*. Tryb ten jest szczególnie przydatny w przypadku przeszukiwania wielu obszernych baz danych. Pewnym utrudnieniem wynikającym z przyjętego rozwiązania jest brak ograniczenia czasowego na otrzymywanie wyników oraz potencjalna możliwość wyszukania tego samego utworu w różnych bazach danych. Rozwiązaniem jest inteligentny moduł prezentacji informacji umożliwiający eliminację powtórzeń, sortowanie wyników wg stopnia podobieństwa oraz systematycznego odświeżania informacji.

W obydwu przedstawionych architekturach istotną rolę odgrywa multimedialny interfejs użytkownika wykorzystujący do realizacji swoich funkcji następujące komponenty:

- moduł wprowadzania różnorodnych zapytań muzycznych wraz z algorytmami ich konwersji do monofonicznego nagrania MIDI,
- interaktywny edytor zapytań muzycznych,
- moduł tworzenia zapytań muzycznych w postaci konturów muzycznych z opcjonalną informacją o interwałach,
- moduł komunikacji z sieciowym środowiskiem agentów (serwerem agencji),
- moduł zbierania wyników wyszukiwania, ich interpretacji oraz wyświetlania w oknie przeglądarki *Internet Explorer*.

W kolejnych punktach pracy przedstawione zostaną zagadnienia związane z tworzeniem zapytań muzycznych oraz algorytmy dopasowania melodii (*melody matching*).

3. MUZYCZNE DANE WEJŚCIOWE

Wprowadzanie zapytań muzycznych do wyszukiwania w multimedialnej bazie danych nie jest procesem tak trywialnym, jak wprowadzanie zapytań tekstowych. Zapytanie muzyczne ma postać wzorca melodycznego, zawierającego z reguły od kilku do kilkunastu dźwięków przynależących, czasem jedynie częściowo, do poszukiwanego utworu. Prawidłowo zaprojektowany interfejs użytkownika musi uwzględnić w tym wypadku dużą różnorodność technik wprowadzania zapytań muzycznych do systemu komputerowego związaną z różnymi preferencjami użytkowników.

W projekcie systemu *MelodyExplorer* założono możliwość wprowadzania zapytań na jeden z poniższych sposobów:

- zagranie melodii na instrumencie komunikującym się za pomocą interfejsu MIDI (również wczytanie nagranego uprzednio pliku MIDI),
- nagranie za pomocą mikrofonu melodii zanuconej (zagwizdanej) lub zagranej na dowolnym nieperkusyjnym instrumencie muzycznym (również wczytanie nagranego uprzednio pliku WAV),

- skanowanie drukowanej partytury (również wczytanie pliku BMP lub TIFF),
- zagranie melodii na klawiaturze komputera,
- graficzne wprowadzenie melodii we wbudowanym edytorze notacji muzycznej bądź modyfikacja wcześniej wprowadzonych zapytań innymi sposobami,

Różnorodność źródeł zapytań dźwiękowych rodzi szereg problemów naukowych z zakresu przetwarzania i rozpoznawania obrazów związanych z koniecznością wydobycia informacji muzycznej z danych dźwiękowych lub graficznych. Zostaną one omówione w kolejnych podrozdziałach.

3.1. Ekstrakcja melodii z muzyki monofonicznej

Jednym z bardziej użytecznych sposobów wprowadzania zapytań muzycznych jest ich bezpośrednie zanucenie, zagwizdanie lub też zagranie na dowolnym instrumencie. Wprowadzenie takiego zapytania do komputera odbywa się w najprostszym przypadku przy użyciu mikrofonu. Z uzyskanego w ten sposób dźwięku monofonicznego należy wydobyć jego podstawowe parametry, tzn. wysokość i czas trwania nut. Zaniedbywane są przy tym amplituda i barwa dźwięku.

Podstawowym problemem w rozpoznawaniu muzyki monofonicznej jest segmentacja sygnału na poszczególne dźwięki i fragmenty ciszy. W ogólności, proces segmentacji można przeprowadzać w dziedzinie czasu lub w dziedzinie częstotliwości. Segmentacja w dziedzinie czasu jest skuteczna jedynie dla prostych sygnałów. Segmentacja w dziedzinie częstotliwości jest podejściem bardziej zaawansowanym, które korzystać może m.in. z analizy mocy sygnału lub HFC (*High Frequency Content*).

Specyficznym rodzajem segmentacji, stosującym podejście mieszane, jest segmentacja w dziedzinie wykrytej częstotliwości podstawowej, badająca zmienność w czasie chwilowej wartości wysokości (częstotliwości podstawowej) dźwięku. Istnieje wiele metod śledzenia wysokości dźwięku. W systemie *MelodyExplorer* zaimplementowano metodę autokorelacyjną [6]. W metodzie tej, dla kolejnych okienek czasowych, wyznaczana jest funkcja autokorelacji zgodnie ze wzorem:

$$\hat{r}(n) = \frac{1}{N} \sum_{k=0}^{N-n-1} x(k) \cdot x(k+n) \tag{3.1}$$

gdzie: N – liczba próbek w oknie,
n – przesunięcie w dziedzinie czasu,
x(k) – badany sygnał.

Pierwsze maksimum lokalne sygnału autokorelacji dla wartości n_b wyznacza wartość częstotliwości podstawowej f_b w oknie wg wzoru:

$$f_b = \frac{f_p}{n_b} \tag{3.2}$$

gdzie: f_p – częstotliwość próbkowania.

Działanie algorytmu autokorelacji przedstawiono na rys. 2, 3 i 4 dla przykładowego nagrania gamy *C-dur* (rys. 2) zanuconej do mikrofonu. Efekt bezpośredniego zastosowania algorytmu autokorelacji z wykorzystaniem okna o szerokości N = 2048 próbek (rys. 3) wskazuje występowanie lokalnych zakłóceń (pików i fluktuacji) o szerokości 1 – 2 okien. Zakłócenia te eliminowane są filtrem medianowym o szerokości 7 okienek (rys. 4). Z uwagi na niedokładność śpiewu człowieka, pewną wrażliwość metody

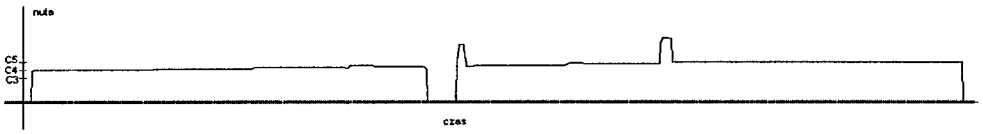
na stany przejściowe między dźwiękami lub ciszę, a także inne czynniki, takie jak złe nastrojenie instrumentu, kolejnym etapem jest usunięcie lokalnych fluktuacji wysokości dźwięku oraz dopasowanie wykrytych częstotliwości do wartości adekwatnych dla danej tonacji.



Rys. 2. Gama C-dur zanucona do mikrofonu – amplituda zmieniająca się w czasie



Rys. 3. Wyjście algorytmu śledzenia dźwięku dla gamy C-dur



Rys. 4. Efekt wygładzania wyjścia algorytmu śledzenia wysokości dźwięku przy użyciu filtru medianowego o rozmiarze okna = 7

3.2. Ekstrakcja melodii z polifonicznych utworów MIDI

W ogólności pliki MIDI opisują muzykę polifoniczną, tzn. dla wielu grających jednocześnie głosów (instrumentów). Informacje dla poszczególnych instrumentów przechowywane są na oddzielnych ścieżkach lub w osobnych kanałach pojedynczych ścieżek. W celu stworzenia monofonicznego zapytania do bazy danych konieczna jest więc ekstrakcja melodii z utworu polifonicznego. Zagadnienie to jest skomplikowane z uwagi na konieczność uwzględnienia teorii percepcji muzyki i do tej pory nie pojawiło się żadne, w pełni satysfakcjonujące rozwiązanie tego problemu.

Poszczególne instrumenty pojawiające się w pliku MIDI mogą odgrywać partie akordowe składające się z kilku jednoczesnych dźwięków. Istnieje zatem możliwość występowania kilku równoczesnych wątków melodycznych, zaś odbierana melodia może przemieszczać się z jednego instrumentu na drugi. Co więcej, w niektórych partiach melodia może w ogóle nie występować, a więc trudno jest stwierdzić, jaka sekwencja dźwięków powinna być użyta do reprezentacji danego fragmentu muzyki. Informacji o melodii nie niosą również dźwięki perkusyjne, które są z reguły pomijane w analizie melodycznej utworu.

W celu wydobycia melodii z utworu muzycznego stosuje się wiele podejść. Z reguły korzystają one z pewnego modelu percepcji akustycznej człowieka i wynikających z niej reguł. Przykładowymi regułami są:
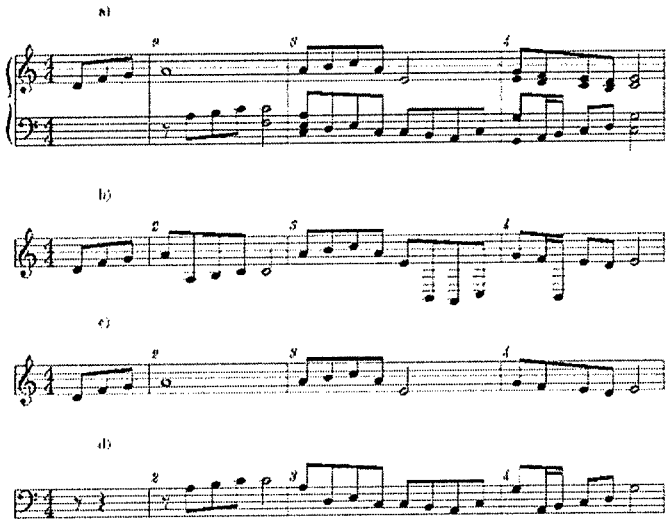
- w utworach muzycznych jako melodia postrzegana jest zazwyczaj partia najwyższa,
- dźwięki położone blisko siebie na skali wysokości z reguły należą do tej samej partii muzycznej.

W innych pracach wykorzystuje się pierwszorzędową entropię predykcyjną, do określenia możliwości przewidywania kolejnych dźwięków w utworze. Dla utworów o niskiej wartości entropii takich, jak np. pewne formy akompaniamentu, istnieje duże prawdopodobieństwo predykcji kolejnych dźwięków melodii za pomocą prostego modelu Markova, z jednym stanem dla każdego dźwięku.

W kilka pracach poruszono problem rozszczepienia muzyki polifonicznej na części przy użyciu reguł opartych na bliskości dźwięków dla zbioru utworów muzycznych o pewnym wspólnym charakterze [7]. Postuluje się tam również zignorowanie instrumentów perkusyjnych oraz zastosowanie kilku prostych heurystyk.

W publikacji [8] zaproponowano trzy różne metody ekstrakcji melodii z pliku MIDI. Metody te zostały zaimplementowane w systemie *MelodyExplorer*, w którym stały się podstawą do dalszych badań. Wszystkie metody przebiegają jednokrotnie strumień dźwięków, aby wybrać dźwięki należące do melodii, ignorując przy tym zdarzenia z kanału 10 instrumentów perkusyjnych. Efekty działania algorytmów dla przykładowego fragmentu muzycznego ilustruje rys. 5.

W algorytmie pierwszym MAT-CHP (*merge all tracks – choose highest pitch*) wszystkie zdarzenia muzyczne łączy się w jeden strumień. W momencie rozpoczęcia wielu dźwięków, wybierany jest zawsze najwyższy a wszystkie pozostałe, rozpoczynające się w czasie jego trwania są usuwane (rys. 5b). W algorytmach drugim GMTwHP (*get mono track with highest pitch*) jako melodia wybierana jest zawartość kanału z najwyższą średnią wysokością nut (rys. 5c). Z kolei, w algorytmie trzecim GMTwHE (*get mono track with highest entropy*) wybierana jest zawartość kanału z najwyższą pierwszorzędową entropią predykcyjną (rys. 5d).



Rys. 5. Przykłady ekstrakcji melodii dla fragmentu utworu muzycznego (opis w tekście)
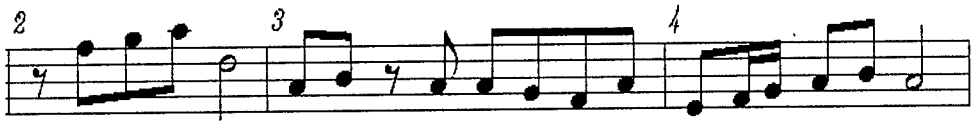
3.3. Ekstrakcja melodii z obrazu partytury

Jednym z możliwych sposobów wprowadzania zapytania muzycznego jest obraz drukowanej partytury. Obraz taki może być wprowadzony do systemu komputerowego za pomocą skanera lub aparatu cyfrowego (*off-line*) lub przy użyciu kamery cyfrowej (*on-*

line). W projekcie *MelodyExplorer* wykorzystano system rozpoznawania notacji muzycznej (*OMR*) *ScoreExplorer* [9]. System ten charakteryzuje się wysoką skutecznością rozpoznawania partytur muzycznych, zarówno monofonicznych jak i polifonicznych. Wymiana danych pomiędzy systemami odbywa się z wykorzystaniem strumienia MIDI.

4. DOPASOWANIE KONTURÓW MELODYCZNYCH

Jak pokazują badania z psychologii muzycznej [10] melodie są rozpoznawane przez człowieka niezależnie od tonacji, w której zostały zagrane lub zaśpiewane. Co więcej, zaniedbanie bezwzględnych wartości interwałów i operowanie wyłącznie na samym ich kierunku, wciąż pozwala na bardzo dobre rozpoznawanie melodii. Notacja konturu melodycznego zaproponowana przez Parsona w 1975 r. [11] pozwala na opisanie dźwięku monofonicznego przy użyciu 4 symboli: '*' – pierwsza nuta, 'D' – opadający interwał (down), 'U' – wznoszący się interwał (up) oraz 'R' (repeat) – powtórzenie wysokości. Przykładowy zapis melodii w postaci jej konturu przedstawiono na rys. 3.



***UUDDUDRDDUDUUUUD**

Rys. 6. Przykładowa melodia i jej kontur.

Kontur muzyczny wydaje się być idealnym sposobem indeksowania baz danych muzycznych i tworzenia do nich zapytań. Jednak w praktycznych zastosowaniach bywa on wzbogacony informacją o interwałach, tonacji lub nawet dokładnym zapisem muzycznym, np. MIDI. Samo dopasowanie zapytań do indeksów w bazie danych może się odbywać w sposób dokładny lub przybliżony. Badania przeprowadzone przez McNaba [3] i Seligę [12] wykazały, że dla bazy 10 000 nagrań długość zapytania powinna wynosić od kilku nut (dla dokładnego dopasowania konturu oraz interwału) do kilkunastu dźwięków przy przybliżonym dopasowywaniu samego konturu.

4.1. Melodia i akompaniament

Aby możliwe było niezawodne wydobycie melodii z polifonicznego utworu, potrzebne jest stwierdzenie, co osoba słuchająca muzyki postrzega jako melodię. Kilka prac, które ukazały się do tej pory, zajmowało się już tym zagadnieniem – badały jak przebiega percepcja grupy dźwięków. Melodia słyszana jest jako dominanta, jeżeli jest grana wyżej od części akompaniamentu. Jednakże, kiedy górne dźwięki są stałe a niższe dźwięki tworzą bardziej interesujący fragment, wtedy to niższe dźwięki słyszane będą jako dominanta. Deutsch [13] prowadzi dyskusje na temat głównych wytycznych i pryncypiów w percepcji grup dźwięków i pokazuje wyniki swoich badań dla percepcji muzyki. Stwierdzono tam również, iż amplituda lub głośność dźwięków jest raczej mało ważna w odbiorze muzyki w porównaniu do bliskości, lecz także może być użyta do grupowania dźwięków.

4.2. Percepcja podobieństwa melodii

W wielu przypadkach, zapytanie muzyczne zostaje dopasowane do wielu melodii. Należy więc opracować miarę podobieństwa utworu do zapytania, pozwalającą na szeregowanie otrzymanych wyników wyszukiwania. Badania dotyczące percepcji muzyki [8] na poniższą klasyfikację podobieństwa utworów (w kolejności malejącego podobieństwa):

- dokładna transpozycja w tej samej tonacji,
- dokładna transpozycja w zbliżonej tonacji,
- dokładna transpozycja w odległej tonacji,
- dokładna transpozycja z wyjątkiem dźwięków zmienionych chromatycznie,
- dokładna transpozycja z wyjątkiem dźwięków zmienionych diatonicznie,
- ten sam kontur i tonalność,
- ten sam kontur, ale atonalny,
- te same wartości nut, ale inny kontur,
- inny kontur, inne wartości nut.

Dla każdej klasy istnieje możliwość stworzenie bardziej precyzyjnej miary ilościowej wskazującej na procent dźwięków prawidłowo rozpoznanych.

Z reguły, trzy pierwsze klasy mogą być połączone w jedną, gdyż dla porównania melodii używa się zazwyczaj interwałów względnych, a człowiek słabo odróżnia tonację muzyczną. Badania nad percepcją muzyki [8] nie ujawniły dotąd potrzeby dokładnego porównywania rytmu lub przebiegu czasowego melodii. Wskazują one natomiast, iż dużo większe znaczenie mają różnice w dźwiękach akcentowanych, co pozwala na przypisanie im wyższej wagi niż dźwiękom nieakcentowanym.

5. APLIKACJA I EKSPERYMENTY

W celu praktycznej weryfikacji i dalszych badań zaprezentowanych algorytmów stworzono system przeglądarki multimedialnej o nazwie *MelodyExplorer*. Został on osadzony w równolegle rozwijanym środowisku sieciowym agencji oraz prostej bazy multimedialnej [4,5,12] (rys. 1). System został napisany w języku C++ (Microsoft *Visual 6.0*) dla środowiska Microsoft Windows. W programie *MelodyExplorer* osadzony został komponent ActiveX przeglądarki Microsoft *Internet Explorer*, który dzięki interfejsowi COM został wykorzystany do obsługi komunikacji internetowej. Rozwiązania takie daje dodatkowe korzyści takie, jak:

- wyświetlanie dowolnie złożonych stron HTML oraz możliwość stosowania dowolnych agentów,
- przezroczysta komunikacja (CGI), szyfrowanie (SSL) i autoryzacja (protokół HTTPS serwer agencji),
- możliwość obsługi dużej liczby formatów danych multimedialnych oraz sprzężenie z zewnętrznymi programami za pomocą technologii ActiveX (np. komponent *Acrobat Reader* do odczytu plików PDF notacji muzycznych przechowywanych w bazie danych muzycznych).

W systemie zaimplementowano 3 algorytmy wydobywania melodii z utworów polifonicznych, opisane w punkcie 3.2. Metody te zostały zaadoptowany do danych pochodzących z różnych źródeł. Przeprowadzone eksperymenty potwierdziły przeciętną

skuteczność algorytmów. W celu polepszenia skuteczności opracowano dodatkowe opcje dla najbardziej obiecującego algorytmu MAT-CHP:

- usuwanie wszystkich zdarzeń perkusyjnych (kanał 10 MIDI)
- usuwanie niepożądanego akompaniamentu, który występując pomiędzy nutami melodii, skraca czas ich trwania. Usuwanie akompaniamentu dotyczy jedynie partii akompaniamentu granych w pewnej odległości od partii melodycznej (w dziedzinie wysokości dźwięku) i może przebiegać następująco:
  - usuwane są wszystkie zdarzenia poniżej pewnej zadanej wysokości
  - usuwane są zdarzenia, których wysokość jest odległa od średniej wysokości poprzedzających je K zdarzeń nutowych co najmniej o zadaną wartość. Początkowa średnia wysokość równa się średniej wysokości wszystkich dźwięków uzyskanych po wyborze najwyższych nut.
- usuwanie pauz – przydatne przy rozpoznawaniu śpiewu oraz po procesie usuwania akompaniamentu.

Tak ulepszony algorytm dawał lepsze wyniki niż oryginalna bazowa metoda przedstawiona w [8]. Subiektywna ocena uzyskanej melodii, poprawiła się o 0,5 – 1 punkt (w pięciopunktowej skali). Eksperymenty pokazały, że melodia lepiej wydobywana jest z utworów muzyki klasycznej niż z muzyki współczesnej.

Do wyznaczenia podobieństwa dwóch konturów melodycznych zastosowano algorytm dopasowania sekwencji znaków znany z programowania dynamicznego. Dzięki konfigurowalnym kosztom dopasowania pojedynczych symboli, możliwe było doświadczalne wyznaczenie optymalnych parametrów tego algorytmu, które pokazane są w poniższej tabeli.

Tablica 5.1

Optymalne koszty zamiany elementów konturu wyznaczone eksperymentalnie

		Symbol źródłowy		
		D	R	U
Symbol docelowy	D	0	3	7
	R	3	0	3
	U	7	3	0

Ponadto: *koszt wstawienia* = 4, *koszt usunięcia* = 4 dla dowolnych symboli w środku łańcucha.

Algorytm tak sparametryzowany bezbłędnie odnajdował podobne melodie, jednocześnie nisko klasyfikując melodie znacząco różniące się od zadanego wzorca.
Zaimplementowane algorytmy rozpoznawania śpiewu i dźwięku instrumentów tradycyjnych charakteryzowały się trafnością (jakością) rzędu od 74% (śpiew) do 80% (instrumenty muzyczne) prawidłowo rozpoznanych dźwięków.

6. ZAKOŃCZENIE

Przeprowadzone eksperymenty wykazały, iż żadne aktualnie znane jednostronne podejście do rozpoznawania śpiewu czy dopasowania melodii nie jest uniwersalne i nie zapewnia wystarczającej jakości wyszukiwania. Lepsze efekty daje zastosowanie metod mieszanych oraz pewnych heurystyk. Niewątpliwe potrzeba dalszych prac i badań, aby omówiony tu system i użyte w nim algorytmy miały jakość przemysłową, jednak wydaje się, że przyjęte założenia oraz podstawy teoretyczne dobrze sprawdzają się w praktyce i są

solidnym fundamentem dla dalszego udoskonalenia algorytmów i ich implementacji. Już obecnie możliwość wyszukiwania danych muzycznych w sposób zorientowany na zawartość stała się faktem.

BIBLIOGRAFIA

[1] Fox E. A., Marchionini G.: *Toward a Worldwide Digital Library*, Communication of the ACM, December 1998.
[2] McNab R. J. at al: *Tune Retrieval in The Multimedia Library*, New Zealand University of Waikato.
[3] McNab R. J. at al: *Towards the digital music library: Tune retrieval from acoustic input*, Proceedings of the ACM Digital Libraries 96, Bethesda, Maryland.
[4] Bruski P., Gardias M.: *System realizacji autoryzowanych usług za pomocą mobilnych agentów do wynajęcia*, praca dyplomowa Politechnika Gdańska WETI-KTP, 2001.
[5] Sopala A., Lautenbach M.: *Konfigurowalny agent klienta protokołu Z39.50*, praca dyplomowa WETI-KTP, 2001.
[6] Yariv Shavit: *Pitch detection using (narrowed) autocorrelation function*, on-line.
[7] Ghias A., Logan J., Chamberlin D., Smith B.: *Query by humming – music information retrieval in an audio database*, ACM Multimedia 1995.
[8] Uitdenbogerd A.L., Zobel J.: *Manipulation of Music for Melody Matching*, Communication of the ACM, August 1998, p. 235 – 240.
[9] Szwoch M: *Rozpoznawanie drukowanej notacji muzycznej z wykorzystaniem bezkontekstowych gramatyk atrybutowych*, praca doktorska, WETI PG, Gdańsk 2002.
[10] Dowling W. J., *Scale and contour: Two components of a theory of memory for melodies*, Psychological Review 85/1978.
[11] Parsons D.: *The Directory of Tunes and Musical Themes*, Spencer Brown, Cambridge 1975
[12] Seliga W.: *Metody formułowania zapytań do wyszukiwania informacji multimedialnej*, praca dyplomowa WETI-KTP, 2001.
[13] Deutsch D.: *Grouping mechanisms in music*, The Psychology of Music 1982, p. 99-134.

MELODY CONTOURS EXTRACTION AND MATCHING FOR MUSICAL DATABASES

Summary

In this paper some algorithms for music data indexing and retrieval from multimedia databases are presented. Various methods of musical queries input are described with appropriate melody extraction algorithms. The *melody contours* are proposed as universal mechanism for music indexing. The prototype system *MelodyExplorer* allows validation of algorithms and carrying on some experiments in the fields of musical data input, melody extraction and melody contour matching. *Melody Explorer* has a very flexible interface that accepts musical data in various forms: MIDI and music files, direct humming or playing MIDI instruments, manual input in built-in music editor or data import from OMR *ScoreExplorer* system. Finally, the results of the experiments with the *MelodyExplorer* system are presented.

Key words: music recognition, melody extraction, melody matching, musical contours, multimedia database.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Piotr Szczuko, Bożena Kostek

Katedra Systemów Multimedialnych, Politechnika Gdańska

BADANIE WEKTORA PARAMETRÓW DO AUTOMATYCZNEGO ROZPOZNAWANIA STYLÓW MUZYCZNYCH

Streszczenie

W referacie przedstawiono badania nad doborem parametrów w wektorze cech, służącego do automatycznego rozpoznawania stylu utworów muzycznych. W celu przeprowadzenia eksperymentów zbudowano bazę danych muzycznych zawierającą fragmenty utworów z kilkuset płyt kompaktowych. Zgromadzone utwory przydzielono do odpowiednich stylów muzycznych, wykorzystując w tym celu format danych zawarty na płytach kompaktowych, służący do opisu płyt CD w Internecie. W procesie automatycznej klasyfikacji obiektów wykorzystano trzy bloki składowe: segmentację sygnałów, parametryzację sygnałów oraz klasyfikację obiektów w oparciu o system decyzyjny oparty o metodę zbiorów przybliżonych. W procesie parametryzacji wykorzystywano opis cech zawarty w standardzie MPEG-7. Uzyskane wyniki wskazują, iż możliwe jest automatyczne rozpoznawanie stylu utworu muzycznego w oparciu o analizowany wektor cech.

1. WPROWADZENIE

Przedstawione w referacie zagadnienia odnoszą się do szeroko pojętej dziedziny *Music Information Retrieval* [1][2]. Opracowanie systemów automatycznego wyszukiwania utworów muzycznych w oparciu o ich reprezentację sygnałową staje się obecnie możliwe. Niektóre jednak zagadnienia pozostają wciąż nierozwiązane. Kluczowym zagadnieniem w rozpoznawaniu sygnałów muzycznych jest ich parametryzacja. Standard MPEG-7 zawiera szereg rozwiązań przeznaczonych do opisu danych multimedialnych [3][4]. Dla danych dźwiękowych oprócz opisu formatu danych, meta danych, standard ten definiuje deskryptory niskiego poziomu wykorzystujące informacje pochodzące z sygnału muzycznego.

W referacie opisano problem przeszukiwania bazy danych muzycznych w oparciu o opis utworów muzycznych w formie wektora parametrów. Na wejście systemu decyzyjnego podawany jest fragment utworu testowego, który został w pierwszej kolejności poddany segmentacji, a następnie parametryzacji w oparciu o parametry zawarte w standardzie MPEG-7. Celem wyszukiwania jest określenie, do jakiego stylu muzycznego dany utwór należy. W trakcie eksperymentów skonstruowano bazę muzyczną zawierającą fragmenty utworów pochodzących z kilkuset płyt kompaktowych. Zgromadzone w bazie utwory zawierają dodatkowo meta opis wg formatu CDDB (Compact Disc Database) [5],

pozwalający na wyszukanie utworu na podstawie danych tekstowych i jednocześnie przydzielenie danego utworu do odpowiedniego stylu muzycznego. Format CDDB definiuje m.in. następujące pola informacyjne:

DISCID – numer identyfikujący dysk w bazie
DTITLE – tytuł płyty
DYEAR – rok nagrania
DGENRE – styl muzyczny
TTITLEnumer – tytuł utworu o wskazanym numerze porządkowym
EXTD – informacja dodatkowa o płycie
EXTTnumer – informacja dodatkowa o utworze

Informacjami kluczowymi w prowadzonych eksperymentach są pola DGENRE, określające gatunek muzyki oraz pola identyfikujące poszczególne nagrania, czyli DTITLE i TTITLEnumer.

W eksperymentach wykorzystano system decyzyjny oparty o metodę zbiorów przybliżonych, przygotowany w Uniwersytecie Warszawskim [6]. Badania prowadzono na wybranym losowo zestawie 155 nagrań: 58 utworów muzyki klasycznej, 70 utworów jazzowych i 27 rockowych.

2. ANALIZA DANYCH DO CELÓW AUTOMATYCZNEJ KLASYFIKACJI MUZYKI

Przygotowane i skatalogowane nagrania zostały poddane analizie w oparciu o zalecenia standardu MPEG-7. Każdy utwór podzielono na segmenty o długości 1 sekundy, które były następnie poddane parametryzacji. Dla każdego segmentu, podzielonego na ramki analizy o długości 30 ms z zakładkowaniem 66%, obliczono następujący wektor parametrów [3]:

- ASE1...ASE26 – *AudioSpectrumEnvelope* (krótkookresowe widmo gęstości mocy) w kolejnych pasmach częstotliwości o szerokościach 1/3 oktawy. Otrzymane wartości zostały zamienione na skalę decybelową, a następnie podzielone przez L2-norm (pierwiastek sumy kwadratów) [3]. Wektor wynikowy dla segmentu uzyskano uśredniając wartości ASE w każdym paśmie po ramkach;
- ASE_mean – średnia wartość parametrów ASE1...ASE26 dla danego segmentu;
- ASE_var – wariancja wartości parametrów ASE1...ASE26;
- ASC_mean – wartość średnia *AudioSpectrumCentroid* po ramkach dla parametryzowanego segmentu;

$$C = \frac{\sum_n \log_2(f(n)/1000) P_x'}{\sum_n P_x'(n)} \tag{2.1}$$

gdzie: P_x' – moc związana z *n-tą* częstotliwością *f(n)*

- ASC_var – wariancja wartości parametrów ASC dla wszystkich ramek parametryzowanego segmentu;

- ASS_mean, ASS_var – wartość średnia i wariancja wartości parametrów *AudioSpectrumSpread*:

$$S = \sqrt{\frac{\sum_n (\log_2(f(n)/1000) - C)^2 P_x'(n)}{\sum_n P_x'(n)}} \tag{2.2}$$

- SFM_mean – *AudioSpecrumFlatness* (stosunek średniej geometrycznej do arytmetycznej współczynników widma gęstości mocy w wybranym paśmie), w każdej ramce parametr ten jest liczony w oddzielnych pasmach częstotliwości, otrzymaną macierz uśredniono po ramkach uzyskując średnie wartości parametrów SFM w każdym paśmie częstotliwości, a następnie obliczono wartość średnią (analogicznie, jak dla ASE);

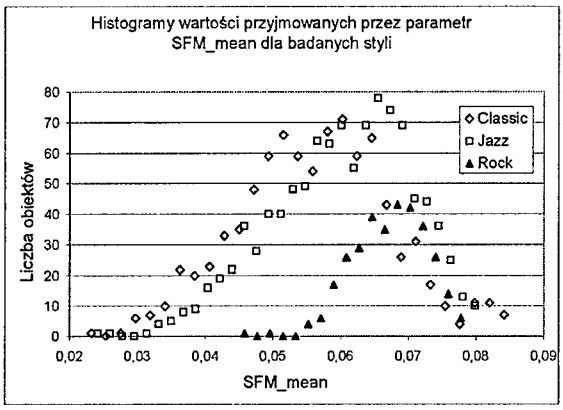
- SFM_var – wariancja uśrednionych po ramkach wartości parametrów SFM w każdym paśmie.

W praktyce do wstępnych badań wykorzystano po 15 segmentów dla każdego utworu, odpowiadających fragmentowi od 1 min 0 s do 1 min i 15 s. Uzyskane dane liczbowe, powiązane z elementami bazy utworów, podzielono na zbiór testowy oraz uczący w stosunku 2 do 3. Zbiór uczący zawierał sparametryzowane pierwsze 9 segmentów z każdego nagrania, zbiór testowy kolejne 6 segmentów.

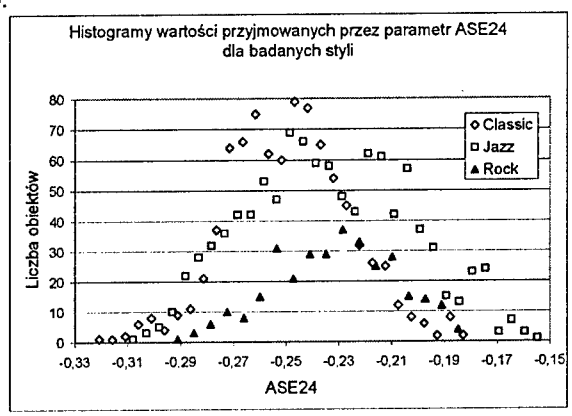
Klasyfikacja i rozróżnienie dźwięków muzycznych jest zagadnieniem wymagającym analizy w przestrzeni wielowymiarowej. Wartości pojedynczych parametrów dla różnych klas najczęściej pokrywają się (rys. 1 - dla czytelności wykreślono tylko wartości środkowe przedziałów histogramów), dwuwymiarowa reprezentacja także nie pozwala na łatwą separację zbiorów obiektów (rys. 2).

W analizie w oparciu o metodę zbiorów przybliżonych wyznaczone zostały reguły wykorzystujące jednocześnie większą liczbę parametrów i pozwalające na rozróżnianie obiektów z różnych klas, a także klasyfikowanie obiektów nieznanych.

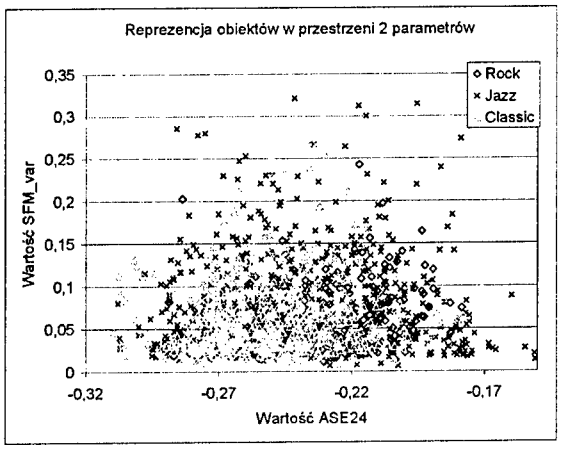
a.



b.



Rys. 1. Porównanie wartości parametrów dla różnych styli: wartość średnia SFM (a), krótkookresowe widmo gęstości mocy dla pasma 24 – ASE24 (b)

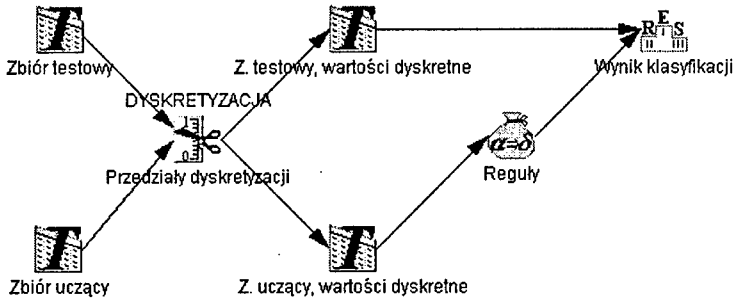


Rys. 2. Dwuwymiarowa reprezentacja obiektów

Przetwarzanie danych (rys. 3) wykonano za pomocą systemu do analizy danych Rough Set Exploration System, pracującego w oparciu o metodę zbiorów przybliżonych [6]. System działa w oparciu o bibliotekę RSES-lib opracowaną w Instytucie Matematyki Uniwersytetu Warszawskiego. System pozwala na obliczenie reguł i reduktów, generalizację reguł, dyskretyzację atrybutów liczbowych, dekompozycję danych, wyszukiwanie regularności wewnątrz danych i inne operacje związane z dziedziną zbiorów przybliżonych [7].

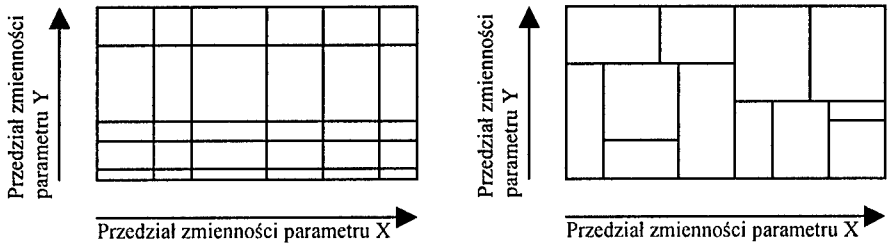
W pierwszym etapie eksperymentów konieczne było wykonanie dyskretyzacji wartości parametrów. Reguły decyzyjne otrzymane na podstawie tabeli decyzyjnej, w której zawarte parametry mają dziedziny ciągłe (atrybuty o wartościach rzeczywistych) są zwykle mało uniwersalne [8], gdyż nie rozpoznają odpowiednio dużej liczby obiektów testowych. Tablice decyzyjne zawierające atrybuty rzeczywiste wymagają dyskretyzacji (skalowanie wartości), czyli zamiany ciągłego zbioru wartości atrybutów na reprezentację w postaci np. przedziałów przynależności. Skalowanie wartości atrybutów polega na wyborze z całego

zakresu zmienności atrybutu szczególnych wartości (cięć), dzielących zbiory wartości na przedziały. W ten sposób liczba różnych wartości w tablicy decyzyjnej maleje do liczby przedziałów. Ważnym zagadnieniem jest tworzenie niesprzecznego i nieredukowalnego zbioru cięć [8].



Rys. 3. Algorytm przetwarzania danych

Nieredukowalny zbiór cięć może być znaleziony przez wielokrotne zastosowanie algorytmu poszukiwania cięcia, które rozróżnia największą liczbę par obiektów, opartego o algorytm *MD-heuristic* [8][9]. W badaniach wykorzystana została dostępna w programie RSES funkcja *Generate Cuts* [6], z zastosowaniem metody *Global*, dokonująca dyskretyzacji w opisany sposób. Metoda globalna tworzy cięcia dla każdego z parametrów osobno, bez uwzględniania zależności między nimi. Przeciwieństwem tej metody jest podejście lokalne (rys. 4).



Rys. 4. Globalna i lokalna dyskretyzacja przestrzeni wartości dwu parametrów

Cięcia wyznaczono w oparciu o zbiór uczący i na ich podstawie dokonano dyskretyzacji wartości parametrów zarówno w zbiorze uczącym, jak i testowym (rys. 3). W tym systemie nie jest konieczne wyznaczanie cięć na podstawie zbioru testowego. Dzięki temu przetwarzać można nieznane dane testowe.

Klasyfikacja nieznanych obiektów ze zbioru testowego przeprowadzana jest za pomocą reguł wyznaczanych na podstawie tablicy decyzyjnej zbioru uczącego (ozn. RSES na rys. 3). W skład reguły wchodzą parametry z tablicy decyzyjnej (zwane reduktami), których wartość pozwala na rozróżnienie pomiędzy elementami zbioru uczącego.

Reguły wyznaczone zostały przy wykorzystaniu algorytmu genetycznego, dostępnego w bibliotece programistycznej RSES-lib. Zaimplementowane wydajne metody genetyczne [6][10], pozwalają na szybkie wyznaczanie określonej z góry liczby reduktów. Dodatkowo wyznaczone reguły są w miarę możliwości najkrótsze, co czyni je użytecznymi w praktycznych zastosowaniach, wymagających szybkości i niskich kosztów obliczeniowych.

Tworzenie reguł, a później klasyfikacja danych testowych, odbywały się dwuetapowo. Celem etapu pierwszego było zebranie wiedzy na temat różnic w parametrach utworów należących do odmiennych styli muzycznych. Dane wprowadzone do systemu RSES w postaci tablicy decyzyjnej miały format, którego fragment pokazano w tabeli 2.1.

Tabela 2.1

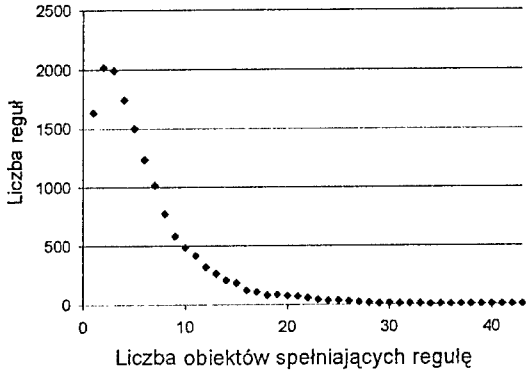
Tablica decyzyjna

ASE1	ASE2	...	ASE26	ASE_mean	ASE_var	ASC_mean	ASC_var	ASS_mean	ASS_var	SFM_mean	SFM_var	Decyzja
-0,18	-0,19		-0,28	-0,19	-0,19	-1,85	0,11	0,92	0,01	0,06	0,06	classical
...												
-0,17	-0,18		-0,35	-0,19	-0,19	-0,61	0,93	1,74	0,13	0,06	0,08	jazz

Wygenerowane reguły wyjściowe mają postać zdania logicznego, uzależniającego decyzję od wartości parametrów (reduktów). Dla przykładu przedstawiono regułę, którą spełniają 43 obiekty ze zbioru uczącego:

[ASE9∈(-∞,-0.15525) ∧ (ASE10∈(-0.16285,+ ∞) ∧ (ASE11∈(-0.16075,+ ∞) ∧
∧ (ASE13∈(-0.18905, +∞) ∧ (ASE26∈(-∞,-0.18935) ∧ (ASE_mean∈(-∞,-0.1614) ∧
∧ (SFM_mean∈(-0.15825,+ ∞)] ⇒ [decyzja=classical]

Łączna liczba wygenerowanych reguł wynosi 15248 o długości od 3 do 12 reduktów. Średnio regułę spełnia 6 obiektów ze zbioru uczącego, maksymalnie 43, minimalnie 1 obiekt (rys. 5).



Rys. 5. Liczba reguł spełnianych przez zadaną ilość obiektów

Nie wszystkie z dostępnych parametrów pojawiają się w regułach, niektóre zostały wykluczone przez algorytm genetyczny, ze względu na brak przydatności w procesie rozróżnienia styli, inne parametry występują natomiast częściej od pozostałych (tabela 2.2).

Tabela 2.2

Udział parametrów w regułach określających styl utworu

Udział parametru w regułach określania stylu	ASE1	ASE2	ASE3	ASE4	ASE5	ASE6	ASE7	ASE8	ASE9	ASE1	ASE11	ASE12	ASE13	ASE14	ASE15	ASE16	ASE17
	38%					39%		37%	39%	40%	33%	38%	23%	37%	34%	26%	

Udział parametru w regułach określania stylu	ASE18	ASE19	ASE2	ASE21	ASE22	ASE23	ASE24	ASE25	ASE26	ASE_mean	ASE_var	ASC_mean	ASC_var	ASS_mean	ASS_var	SFM_mean	SFM_var
			36%				49%		36%	37%			36%			46%	48%

Etap drugi – testowy, przeprowadzono kilkukrotnie, analizując za każdym razem utwory należące do innego stylu. Celem było stwierdzenie, jakie parametry pozwalają na rozróżnienie elementów wewnątrz grupy, czyli utworów w tym samym stylu. Tablica decyzyjna na wejściu systemu miała postać analogiczną do tablicy etapu pierwszego, kolumna „Decyzja" zawierała jednak tytuły poszczególnych utworów.

W następniku implikacji wyznaczonych reguł znajdują się konkretne odwołania do tytułu utworu:

[ASE11$\in$(-∞;-0,1513) $\wedge$ ASE13$\in$(-0,15375;+∞) $\wedge$ ASE24$\in$(-0,23065;+∞) $\wedge$
$\wedge$ ASE_mean$\in$(-∞;-0,18675) $\wedge$ SFM_mean$\in$(-0,18675;+∞)] $\Rightarrow$ [decyzja=Firesofeden]

W regułach określających utwór dla różnych styli biorą występują odmienne zestawy parametrów z różnym udziałem procentowym (tabela 2.3).

Tabela 2.3

Udział parametrów w regułach określających utwór wewnątrz danego stylu

	ASE1	ASE2	ASE3	ASE4	ASE5	ASE6	ASE7	ASE8	ASE9	ASE1	ASE11	ASE12	ASE13	ASE14	ASE15	ASE16	ASE17
utwór rock		35%	39%	32%		41%			47%	44%	46%		46%				
utwór classic	32%		15%	29%	33%		44%			44%	40%	40%	38%	43%			
utwór jazz			39%			42%		39%	42%	41%	59%		25%			35%	

	ASE18	ASE19	ASE2	ASE21	ASE22	ASE23	ASE24	ASE25	ASE26	ASE_mean	ASE_var	ASC_mean	ASC_var	ASS_mean	ASS_var	SFM_mean	SFM_var
utwór rock		48%					49%			55%			59%			52%	
utwór classic			37%		14%		32%			36%				44%	35%	42%	45%
utwór jazz	38%						43%	53%				36%	37%		39%		44%

3. WYNIKI KLASYFIKACJI UTWORÓW MUZYCZNYCH

Wyszukiwanie utworu w bazie odbywa się w sposób hierarchiczny, w oparciu o taksonomię muzyczną. W pierwszej kolejności wyszukiwany plik muzyczny jest przydzielany do odpowiedniego rodzaju muzyki, przy wykorzystaniu reguł zgromadzonych w etapie analizy danych utworów z różnych styli muzycznych. Jeśli parametry nieznanego obiektu spełniają warunki stawiane przez więcej niż jedną regułę, ostateczna decyzja podejmowana jest metodą większościową. Po wybraniu stylu następuje sprawdzenie reguł niższego poziomu wewnątrz tego stylu. Podejmowana jest próba znalezienia właściwego utworu. W tabeli 3.1 przedstawiono trafność rozpoznawania stylu utworów zawartych w bazie testowej. Na podstawie uzyskanych wyników można stwierdzić, że parametry zawarte w wektorze cech pozwalają na rozróżnienie stylu muzycznego, nawet przy liczbie przykładów.

Tabela 3.1

Trafność rozpoznawania stylu utworu

	jazz	classical	rock	ilość obiektów	trafność
jazz	432	82	59	573	0,754
classical	73	386	15	474	0,814
rock	32	7	267	306	0,873

Uzyskana średnia trafność rozpoznawania utworów wewnątrz stylu rockowego wynosi 35%, dla muzyki klasycznej 26%, dla jazzu 37%, co oznacza, że w celu rozpoznawania utworów wewnątrz danego stylu należy uzupełnić wektor cech o dodatkowe parametry.

4. PODSUMOWANIE

Względnie wysoka trafność rozpoznania w pierwszym etapie, dowodzi skuteczności zaproponowanej metody do określania stylu utworu muzycznego. Analizowany zestaw parametrów przyjęty za standardem MPEG-7, dobrze nadaje się do opisu różnic pomiędzy gatunkami muzycznymi. Zastosowane parametry nie sprawdzają się jednak w zadaniu rozróżniania poszczególnych utworów, stąd istnieje potrzeba uzupełnienia wektora cech o dodatkowe parametry.

Podziękowanie
Opisane prace badawcze były dofinansowane w ramach grantu 4 T11D 014 22 KBN.

BIBLIOGRAFIA

[1] Downie J.S.: *Music information retrieval*, Annual Review of Information Science and Technology 37, NJ: Information Today, 2003, ch. 7, 295-340.
[2] Kostek B.: Soft Computing in Acoustics, Physica Verlag, Heidelberg, New York, 1999.
[3] ISO/IEC FDIS 15938-4:2001(E), *Information Technology — Multimedia Content Description Interface — Part 4*
[4] Lindsay A.T., Herre J.: MPEG-7 and MPEG-7 Audio – An Overview, J. Audio Eng. Soc., vol. 49, 7/8, pp. 589-594, 2001.
[5] http://www.gracenote.com/music/
[6] http://logic.mimuw.edu.pl/~rses/
[7] Bazan J.: *Metody wnioskowań aproksymacyjnych dla syntezy algorytmów decyzyjnych*, Ph. D. Thesis, Warsaw University, 1998, 84-86.
[8] Nguyen S.: *Discretization of real value attributes. Boolean reasoning approach.* Ph. D. thesis, Warsaw University, 1997.
[9] Nguyen S., Skowron A.: *Quantization of real value attributes: Rough set and boolean reasoning approach.* Bulletin of International Rough Set Society 1/1, 1997, s. 5-16.
[10] Wróblewski J.: *Finding minimal reducts using genetic algorithms.* Proc. of the Second Annual Join Conference on Information Sciences, September 28-October 1, 1995, Wrightsville Beach, NC, 186-189.

FEATURE VECTOR SELECTION FOR AUTOMATIC MUSICAL STYLE CLASSIFICATION

Summary

In this paper an analysis of parameter effectiveness contained in the feature vector for automatic classification of a musical style is described. For the purpose of experiments a database was constructed including fragments of music from approximately 500 compact discs representing various categories of music. Audio files were gathered according to their styles, based on metadata description compatible to the format of the widely-used CDDB database available in the Internet. The process of automatic classification of musical styles consisted in three blocks: signal segmentation, parametrization and classification employing rough set-based decision system. Selected parameters included in the MPEG-7 standard were used in the parametrization process. The experimental results show that based on the extracted features from music it is possible to recognize a musical style of an audio file.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Wioleta Szwoch

Katedra Inżynierii Wiedzy, Politechnika Gdańska

WYSZUKIWANIE OBRAZÓW W MULTIMEDIALNYCH BAZACH DANYCH

Streszczenie

W pracy przedstawiono problem wyszukiwania obrazów w multimedialnych bazach danych. Wskazano istniejące w tej dziedzinie problemy oraz metody ich rozwiązywania. Pokazano ogólną koncepcję systemu zapytań do multimedialnej bazy danych. Wskazano na wagę problemu wyboru odpowiednich cech obrazu oraz miar podobieństwa. Poruszono problem indeksowania danych. Przedstawiono kilka wybranych algorytmów stosowanych w wyszukiwaniu obrazów. Zaprezentowano prototypowy system wyszukiwania obrazów oraz wstępne wyniki badań wykonanych przy jego użyciu.

1. WSTĘP

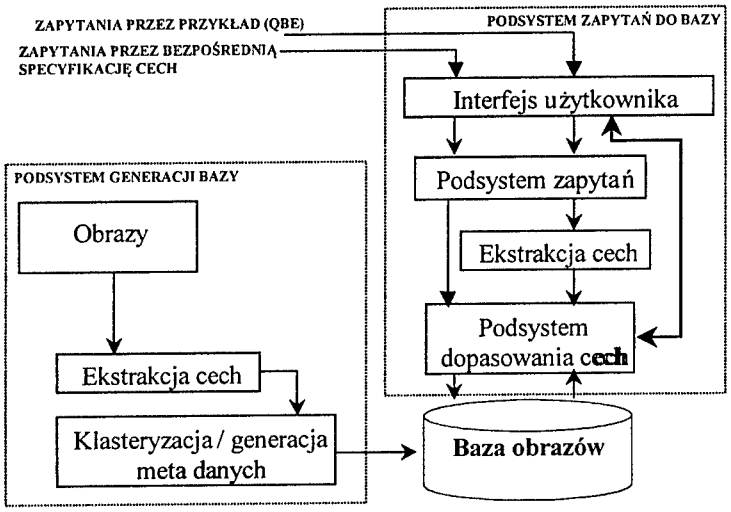
We współczesnym świecie rośnie znaczenie możliwości odnalezienia interesującej nas informacji i szybkiego do niej dostępu. Ponieważ dane przechowywane w bibliotekach cyfrowych mogą być nie tylko informacją tekstową, ale również multimedialną, powstaje problem wyszukiwania informacji tego rodzaju. Obecnie obserwuje się dynamiczny rozwój prac nad stworzeniem wydajnych narzędzi dostępu (składowania, zapytań i wyszukiwania) do multimedialnych materiałów, takich jak np. obrazy, przy użyciu ich specyficznych cech. Systemy takie nazywane są systemami CBIR (*Content Base Image Retrieval*) [1]. Na rys. 1. przedstawiono typowy schemat takiego systemu.

W systemie CBIR można wyróżnić podsystem generacji bazy oraz podsystem zapytań. W podsystemie generacji wydobyte z obrazu cechy służą do wygenerowania indeksów, które będą użyteczne w podsystemie zapytań. Podsystem ten wyszukuje dane porównując cechy obrazu zapytania z cechami obrazów z bazy. System tworzenia zapytań musi być przyjazny użytkownikowi.

W systemach CBIR wyróżnia się następujące rodzaje zapytań:

- przez wybór (QPE – *Query by Pictorial Example*), w którym do wskazania cech szukanych danych wykorzystywane są predefiniowane piktogramy,
- przez przykład (QBE – *Query By Example*), w którym dokonuje się wyboru obrazu zapytania spośród reprezentatywnych wzorców obrazów z bazy; jest to najczęściej wykorzystywany rodzaj zapytań

– wyszukiwanie pseudosemantyczne, w którym do wskazania cech wykorzystywane są opisy tekstowe generowane przez system.



Rys.1. Schemat systemu wyszukiwania danych multimedialnych (*CBIR*)

2. INDEKSOWANIE DANYCH

Efektywne wyszukiwanie informacji w bazie możliwe jest dzięki zaindeksowaniu znajdujących się w niej danych. Z uwagi na ich dużą liczbę istotne jest, by w odpowiedzi na zapytanie otrzymać jak najdokładniejszą listę obrazów podobnych do zapytania. W idealnej sytuacji lista powinna obejmować wyłącznie obrazy pasujące do zapytania. Wynika stąd, że szczególnie istotną rolę w systemach wyszukiwania informacji odgrywa właściwe opisanie i indeksowanie danych.

2.1. Metodologia

Można wyróżnić 3 podstawowe podejścia do opisywania (indeksowania) danych [1]. Najprostszym, z algorytmicznego punktu widzenia, podejściem jest *ręczna kategoryzacja* danych, tzn. opisanie ich przez szereg cech, przynależących do predefiniowanych kategorii. Podejście to nie pozwala na zadawanie zapytań przez przykład.

Z kolei najtrudniejszym podejściem jest *wydobycie z danych informacji* w nich zawartych i użycie ich jako atrybutów opisujących dane w bazie. Podejście to korzysta z zaawansowanych algorytmów rozpoznawania obrazów oraz baz wiedzy i reguł logicznych.

Podejściem pośrednim jest *automatyczna indeksacja*, stanowiąca próbę opisania danych poprzez zbiór charakteryzujących je cech. Tak utworzony wektor opisuje dane, a zdefiniowana funkcja odległości umożliwia znajdowanie danych o podobnych cechach. Podstawowymi zaletami automatycznej indeksacji jest niska złożoność stosowanych w niej algorytmów, krótki czas wprowadzania i wyszukiwania danych oraz możliwość zadawania zapytań przez przykład.

Charakterystyczne cechy metod indeksacji przedstawiono w tabeli 3.1.

Tablica 3.1

Metodologie indeksacji

	Kategoryzacja danych	Wydobycie informacji	Automatyczna indeksacja
Wyniki	+Dokładne	+Dokładne	-Zgrubne
Złożoność algorytmu	Podstawowa	-Bardzo wysoka	+Niska
Zapytania	Tekstowe	Tekstowe/Przykład	Przykład
Automatyzacja	-Przetwarzanie ręczne	Automatycznie	Automatycznie
Czas:			
wprowadzania	-Długi	-Długi	+Krótki
wyszukiwania	+Krótki	-Długi	+Krótki

U podstaw automatycznej indeksacji obrazów leży teza, iż dla każdego zbioru obrazów istnieje zbiór cech oraz miara odległości takie, że możliwe jest skuteczne wyszukiwanie obrazów podobnych do obrazu zapytania.

Podstawowymi problemami są tutaj:
- znalezienie właściwych cech reprezentujących obrazy
- wybranie najlepszej miary odległości obrazów
- oszacowanie skuteczności wybranych cech i miary odległości

2.2. Cechy obrazów i rodzaje szukania

Wybór cech używanych do opisywania obrazów ma kluczowe znaczenie dla procesu wydobywania informacji – zarówno dla jego dokładności jak i szybkości. W ogólności można wyróżnić trzy poziomy wydobywanych cech [2]:

- primitywne (*primitive features*), takie jak kolor, tekstura czy kształt oraz histogramy wybranych cech. Przykładowe zapytanie może mieć wtedy postać: „znajdź wszystkie obrazy zawierające żółte lub niebieskie gwiazdy zgrupowane w koło";
- logiczne (*logical features*), wymagające kilku stopni wnioskowania o podobieństwie obiektów przedstawionych na obrazie. Przykładowym zapytaniem może być w tym przypadku: "znajdź obrazy pociągu pasażerskiego jadącego przez most";
- abstrakcyjne (*abstract attributes*), wymagające złożonego wnioskowania o ważności opisywanych obiektów czy scen (np. "znajdź obrazy ilustrujące urodziny").

Obecnie stosowane są systemy wykorzystujące głównie cechy prymitywne. Najczęściej stosowane są takie cechy jak: kolor, kształt, tekstura oraz histogramy. Eksperymentalne systemy CBIR próbują wykorzystywać cechy logiczne.

Wyszukiwanie obrazów w multimedialnej bazie można podzielić na:
- wyszukiwanie celu (konkretny obraz)
- wyszukiwanie kategorii (jeden lub kilka obrazów z grupy)
- wyszukiwanie swobodne (wszystkie obrazy o zadanych cechach).

2.3. Miary porównania obrazów

Przy porównywaniu obrazu zapytania z obrazami z bazy wykorzystuje się najczęściej rozmaite testy statystyczne oraz miary odległości. Jednymi z najprostszych, a jednocześnie dosyć często stosowanych są:

miara odległości Euklidesa

$$L_E(x,y) = \left(\sum_i |x_i - y_i|^2 \right)^{1/2} \tag{2.1}$$

oraz miara Minkowskiego:

$$L_M(x,y) = \sum_i |x_i - y_i| \tag{2.2}$$

Wykorzystywane są również miary odległości: Chamfera, Hausdorffa, Mahalanobisa , Canberra, Czekanowskiego i inne [1,2].

Aby poprawić jakość wyszukiwania obrazów, często stosuje się w systemach sprzężenie zwrotne. Użytkownik określa, w jakim stopniu odnalezione przez system obrazy odpowiadają jego oczekiwaniom, a system korzystając z informacji zwrotnej uczy się poprawnych odpowiedzi. Po wykonaniu porównań cech obrazy są sortowane i pokazywane począwszy od tych, najbardziej podobnych do obrazu zapytania.

Obok metod statystycznych do porównywania obrazów równie często stosuje się systemy decyzyjne oparte na algorytmach uczących się.

3. METODY WYSZUKIWANIA OBRAZÓW

Zagadnienie wyszukiwania danych multimedialnych rodzi wiele problemów. Oprócz problemów związanych z samą organizacją bazy danych oraz z charakterem i rozmiarem zawartych w niej danych multimedialnych, wyróżnić można zagadnienia związane z indeksowaniem i porównywaniem danych oraz prezentacją wyników. Przykładowo, z uwagi na dużą liczbę danych zawartych w bazie, istotne jest, by w odpowiedzi na zapytanie otrzymać wyłącznie pasujące obrazy a także, by żadnego pasującego obrazu nie pominąć. W przypadku konieczności wyszukiwania danych w czasie rzeczywistym (*on line*), konieczne jest stosowania szybkich algorytmów.

Należy też zwrócić uwagę, że aby znaleźć dobre dopasowanie, trzeba uporać się ze zmianami punktu widzenia, okluzjami i zmieniającą się rozdzielczością obrazu. Stosowane metody powinny być niewrażliwe na szumy oraz niezmienne względem obrotu, przesunięcia i skali. Istotne znaczenie ma również uniezależnienie się od wpływu warunków oświetlenia zdjęcia. Przy sprawdzaniu poprawności działania metody i jej skuteczności problemem może być również subiektywizm przy określaniu podobieństwa między obrazami.

W kolejnych podrozdziałach przedstawiono wybrane metody wyszukiwania obrazów.

3.1. Przecięcie histogramów

Histogramy kolorów są jedną z najintensywniej wykorzystywanych cech w systemach wyszukiwania obrazów. Metody wykorzystujące histogramy różnią się sposobem tworzenia oraz porównywania histogramów. Jednym z pierwszych podejść było zastosowanie tzw. *przecięcia histogramów* zaproponowane przez Swaina i Ballarda [3]. Ta prosta koncepcyjnie i w realizacji metoda, stanowi podstawę wielu innych prac badawczych i jest traktowana jako punkt odniesienia do uzyskanych w nich wyników.

Przecięcie histogramów wyrażane jest wzorem (3.1), który opisuje rozmiar części wspólnej histogramów obrazu I oraz zapytania Q, znormalizowanej względem liczby pikseli histogramu zapytania. Wyższa wartość przecięcia (maksymalnie 1.0) oznacza lepsze

dopasowanie. Przyjmując, iż każdy prążek histogramu oznacza cechę, przecięcie histogramów może być traktowane jako klasyczne rozpoznawanie obrazów. Opisany przez cechy obiekt, staje się wtedy punktem w N-wymiarowej przestrzeni cech, gdzie N jest liczbą prążków histogramu. Przecięcie histogramów zachowuje wszystkie cechy niezmiennicze histogramów, a więc: niezmienność względem translacji i obrotu oraz niewielką zależność od kąta widzenia.

$$D_{\text{int}} = \frac{\sum_{i=1}^{N} \min(H_Q(i), H_I(i))}{\sum_{i=1}^{N} H_Q(i)} \tag{3.1}$$

gdzie: Q – obraz zapytania,
I – obraz bazy,
H – histogram,
N – liczba prążków histogramu.

Wielu badaczy modyfikuje podstawowy algorytm przekształcając histogram tak, by zredukować liczbę znaczących prążków. Redukcja taka znacząco skraca czas obliczeń, a ponadto uniezależnia porównywanie od drobnych zaburzeń i niewielkich zmian oświetlenia obrazu. Przykładowo, Mehre i inni [4] przedstawili metodę odległości oraz metodę tablicy kolorów odniesienia. Skorzystali oni z faktu, że bardzo często na obrazach istnieje wiele jednolitych płaszczyzn, co pozwala wybrać do porównywania obrazów tylko najistotniejsze prążki histogramów. W praktyce, liczbę znaczących prążków udaje się ograniczyć do ok. 10-30.

3.2. Inne metody

W literaturze można spotkać wiele innych metod wykorzystywanych do wyszukiwania obrazów [1,5]. Wśród metod wykorzystujących kształt występują zarówno techniki bazujące na konturze jak i na całym obszarze kształtu. Do zgrubnego ograniczenia zbioru poszukiwań mogą być wykorzystane proste *deskryptory kształtu* takie, jak np. pole powierzchni, kulistość, centryczność, czy wypukłość. Dopasowanie „punkt-do punktu" (każdy punkt kształtu traktowany jest jako cecha) używa do porównania kształtów *odległości Hausdorffa*. Metoda wykorzystująca *sygnaturę kształtu* nie nadaje się do zastosowań *on line*, jest też wrażliwa na przekształcenia obrazu.

W wyszukiwaniu obrazów wg kształtów szeroko stosuje się *momenty geometryczne* oraz bazujące na nich *niezmienniki momentowe*, które są niezmienne względem obrotu, przesunięcia i skali oraz w pewnym stopniu niewrażliwe na szumy. Metoda *dopasowania elastycznego* deformuje wzorzec, aby dopasować go do analizowanego kształtu. Wykorzystuje się również często analizę w dziedzinie widmowej używając *deskryptorów Fouriera* oraz *falkowych*. Kształty można również porównywać wykorzystując reprezentację kształtu przy pomocy *kodu łańcuchowego*, dekompozycję na *wielokąty* czy też dekompozycję *krzywymi gładkimi*. Wyszukiwanie obrazów przy użyciu kształtów może też być wykonywane przy użyciu *analizy składniowej* i odpowiednio zdefiniowanej gramatyki.

Decyzja o trafności wyszukiwania podejmowana jest w oparciu o metody statystyczne. Równie często stosuje się systemy decyzyjne oparte na algorytmach uczących się.

4. SYSTEMY WYSZUKIWANIA OBRAZÓW

Na świecie istnieje kilkanaście systemów wykorzystujących różnorodne metody do wyszukiwania obrazów. W systemie *PicToSeek* [6] wykorzystywane są histogramy koloru i kształtu, a wyszukiwanie odbywa się bez wstępnej indeksacji. W systemie *Blobworld* [7], w obrazie wyodrębniane są spójne regiony, które mogą odpowiadać obiektom. Można zadawać proste pytania, chcąc znaleźć obrazy podobne do wybranego regionu, lub tworzyć zapytania złożone, używając innych regionów. Brunelli i Mich stworzyli system wyszukiwania obrazów *Compass* oparty o wykorzystanie histogramów jako deskryptorów obrazu [8-10]. W celu poprawy jakości wyszukiwania użyli sprzężenia zwrotnego z negatywnymi przykładami. W sytuacji, gdy system wyszukuje obrazy nieistotne z punktu widzenia użytkownika, wskazanie ich jako przykładów negatywnych poprawia efektywność systemu. W systemie zaimplementowano również możliwość przeglądania bazy, oferując użytkownikowi mapę. Baza podzielona jest na grupy obrazów podobnych, a każda grupa reprezentowana jest na mapie przez obraz kluczowy. Zapytania podawane są do bazy przez przykład. Można wyróżnić także kilka innych systemów wyszukiwania obrazów: *WebSeek*, *QBIC*, *C-BIRD*, *Chabot*, *MARS* i inne.



Rys.2 Testowa baza danych obrazów. Kółkiem oznaczono zdjęcie, które służy jako przykład do wyszukiwania(QBE). Następujące po nim zdjęcia: 1) w skali szarości, 2) ze zmianą jasności i nasycenia kolorów, 3) po zastosowaniu efektu „OilPaint", 4) obrócone, 5) zaszumione, 6) ze zmniejszoną głębią kolorów, 7) ze zmienionym kontrastem i jasnością.

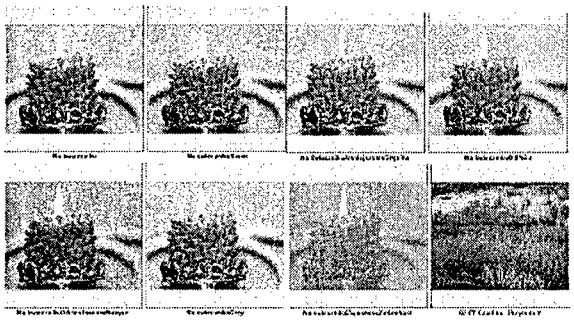
System *ImageExplorer*

W Katedrze Inżynierii Wiedzy rozpoczęto prace nad stworzeniem badawczego systemu wyszukiwania obrazów, który umożliwiałby implementację i weryfikację skuteczności wybranych algorytmów porównywania obrazów. System *ImageExplorer* stworzony został w środowisku Microsoft Visual 6.0 i przeznaczony jest dla systemu operacyjnego Windows. *ImageExplorer* charakteryzuje się przyjaznym interfejsem użytkownika, zbliżonym do popularnych systemów zarządzania albumami zdjęć i zapewnia zbliżoną funkcjonalność odnośnie przeprowadzenia typowych operacji zarządzania obrazami graficznymi. Zaimplementowany w systemie system pomocy (*help*, *wizards*) umożliwia przeprowadzenie użytkownika przez kolejne etapy tworzenia bazy danych, a następnie wyszukiwania w niej obrazów.

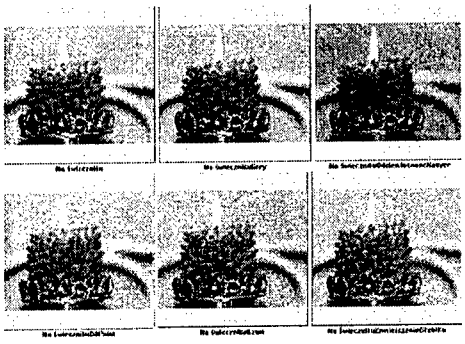
ImageExplorer skonstruowany został według schematu przedstawionego na rys. 1. Część systemu odpowiedzialna za tworzenie bazy danych umożliwia wybór zdjęć w popularnych formatach graficznych (BMP, TIFF, JFIF-JPEG). Z uwagi na potencjalnie różne wymiary zdjęć przyjęto, iż zarówno algorytmy indeksujące jak i rozpoznające będą operowały na obrazach zmniejszonych do wybranego przez użytkownika rozmiaru (miniaturkach). Podejście takie uniezależnia w pewnym stopniu wyniki wyszukiwania od oryginalnych rozmiarów zarówno obrazów w bazie danych jak i obrazów zapytań. Dodatkowymi zaletami są: znaczne ograniczenie czasu indeksowania obrazów i operacji wyszukiwania, a także możliwość wykorzystania miniaturek do przeglądania bazy oraz wyników wyszukiwania. Przeprowadzone badania wykazały, iż najlepsze wyniki osiągane są dla miniaturek kwadratowych o rozmiarze od 64 do 128 pikseli.

Dla każdego obrazu umieszczonego w bazie wyznaczane są cechy (indeksy) służące do jego wyszukiwania. Wektory cech wyznaczane są na podstawie miniaturki obrazu, oddzielnie dla każdej klasy zaimplementowanych metod. W pierwszej fazie badań zrealizowane zostały dwie metody porównywania obrazów, pierwsza porównująca obrazy piksel po pikselu oraz druga, bazująca na deskryptorach Fouriera. W obydwu przypadkach, jako miarę odległości pomiędzy wektorami cech porównywanych obrazów przyjęto odległość Euklidesa (wzór 2.1).

a)



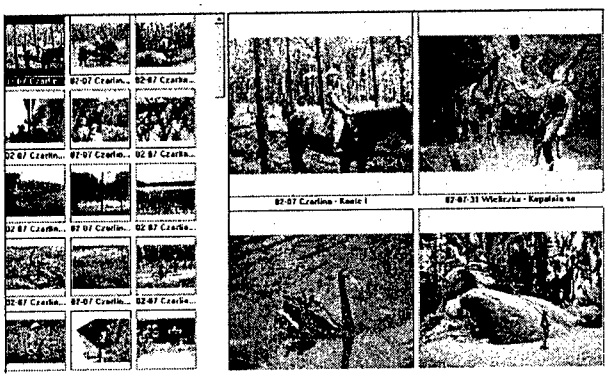
b)



Rys.3. Przykładowe efekty wyszukiwania w bazie (zapytaniem jest obraz „Na świeczniku")

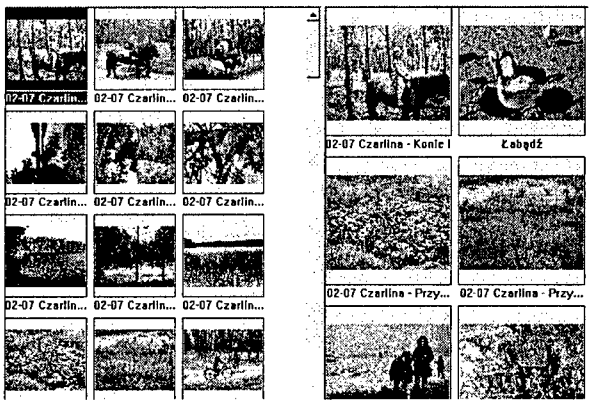
Pierwsze badania przeprowadzone przy pomocy systemu *ImageExplorer* objęły algorytm bazujący na deskryptorach Fouriera. Baza testowa obejmowała kilkadziesiąt fotografii o różnej tematyce (rys. 2). Pierwsze testy objęły problem wyszukiwania w bazie obrazów, będących przekształconymi obrazami zapytania (obraz „Na świeczniku" zaznaczony na rys.2). Przykładowe wyniki wyszukiwania przedstawiono na rys.3a.

Wstępne efekty badań wydają się być zachęcające, wskazując na dość wysoką odporność deskryptorów Fouriera na zmianę jasności i nasycenia kolorów, zaszumienie oraz lokalne zniekształcenia geometryczne. Niestety, brak odporności na obrót oraz trudności z jednoznacznym określeniem progu podobieństwa obrazów, mogą znacząco ograniczyć zastosowania tej metody. Również próby znalezienia przy jej użyciu obrazów o podobnej zawartości (np. konie) zakończyły się jedynie połowicznym sukcesem (rys.4).



Rys.4. Przykładowe efekty wyszukiwania w bazie metodą opartą na deskryptorach Fouriera

Kolejne badania objęły metodę porównującą obrazy piksel po pikselu. Metoda ta teoretycznie nadaje się wyłącznie do wyszukiwania obrazów bardzo podobnych do obrazu zapytania, nie zezwalając na większe odchylenia w położeniu obiektów przedstawionych na obrazach. Tezę tę, potwierdzają wyniki uzyskane dla obrazów skorelowanych (Rys. 3b) oraz niezależnych (rys.5). Uzyskane wyniki wykluczają tę metodę (i jej ewentualne modyfikacje) z dalszych badań.



Rys.5. Przykładowe efekty wyszukiwania w bazie metodą porównywania piksel po pikselu

5. ZAKOŃCZENIE

W pracy przedstawiono problem wyszukiwania obrazów w multimedialnych bazach danych. Wskazano na istniejące podejścia, metody, systemy i pojawiające się problemy. Przedstawiono również wyniki pierwszych badań z prototypowym programem *ImageExplorer*. Tworzony system umożliwi dokładne zbadanie właściwości poszczególnych algorytmów CBIR oraz możliwości ich zastosowania w bibliotekach cyfrowych. Planowane badania nad porównywaniem obrazów przy użyciu cech prostych obejmą zaimplementowany już algorytm bazujący na deskryptorach Fouriera (por. 4) oraz algorytmy bazujące na histogramach (por. 3.1).

Przeprowadzone badania potwierdzają, iż algorytmy wyszukiwania obrazów bazujące na cechach prostych nie mogą osiągnąć skuteczności algorytmów dokonujących analizy semantycznej obrazów (por. rys. 4 i 5). Jest to oczywiste, gdyż sama informacja o podobnej charakterystyce obrazów, czy to barwnej, czy też związanej z innymi cechami, nie gwarantuje, że obrazy będą posiadały podobną zawartość. Mimo to, algorytmy te mogą stać się częścią składową bardziej zaawansowanych algorytmów, wykrywając np. podobieństwa pomiędzy fragmentami obrazów. Ponieważ algorytmy tego typu charakteryzują się dużą szybkością działania, mogą być one wykorzystywane do wstępnej selekcji danych w algorytmach typu *data mining*.

BIBLIOGRAFIA

[1] Antani S., Kasturi R., Jain R.: *A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video*, Pattern Recognition 35, 2002 s.945-965.
[2] Eakins J.P.: *Towards intelligent image retrieval*, Pattern Recognition 35, 2002 s.3-14.
[3] Swain M.J., Ballard D.H.: *Color indexing*, Int.J.Comput.Vision 7 (1), 1991 s.11-32.
[4] Mehrte B., Kankanhalli M., Narasimhalu A., Chang Man G.: *Color matching for image retrieval* Pattern Recognition Letters 16, 1995.
[5] Dengsheng Zhang, Guojun Lu: *Review of shape representation and description techniques*, Pattern Recognition 37, January 2004.
[6] System *PicToSeek*: http://www.wins.uva.nl/research/isis/zomax/.

[7] System *BlobWorld*: http://elib.cs.berkeley.edu/photos/blobworld.
[8] Brunelli R., Mich O.: *Image retrieval by examples*, IEEE Trans. on Multimedia vol.2 no3 September 2000.
[9] Brunelli R., Mich O.: *Histograms analysis for image retrieval*, Pattern Recognition 34, 2001.
[10] System *Compass*: http://compass.itc.it/papers/05_06HTML/INDEX.HTM.

IMAGE RETRIEVAL FROM MULTIMEDIA DATABASES

Summary

In this paper the problem of image retrieval from multimedia databases is described. The main problems in this field and the existing methods of their solving are pointed out. The general conception of multimedia database's query system is presented as well as problem of data indexing. Also, several retrieval algorithms are introduced. Finally, the prototype retrieval system is presented with some initial research done with its help.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Mariusz Szwoch, Wioleta Szwoch

Katedra Inżynierii Wiedzy, Politechnika Gdańska

PRZETWARZANIE WSTĘPNE I SEGMENTACJA ZNISZCZONYCH DOKUMENTÓW ARCHIWALNYCH

Streszczenie

Automatyczne rozpoznawanie maszynopisów jest celem międzynarodowego projektu *Memorial*. Artykuł ten podejmuje temat wstępnego przetwarzania i segmentacji skanowanych dokumentów archiwalnych w celu wyznaczenia rzeczywistych położeń regionów zawierających tekst. Trafne wyznaczenie tych regionów jest czynnikiem warunkującym skuteczność dalszych etapów rozpoznawania zawartości dokumentu.

1. WPROWADZENIE

Strategicznym celem międzynarodowego projektu 5PR *Memorial* jest opracowanie metody tworzenia wirtualnych archiwów wykorzystujących dokumenty znajdujące się w tradycyjnie rozumianych bibliotekach, archiwach, muzeach i urzędach [1]. Cel ten osiągany jest poprzez rozwój modeli danych przeznaczonych do przechowywania informacji, rozwój rozproszonych usług wirtualnych służących nawigacji i poszukiwaniu informacji, a także poprzez zautomatyzowanie procesu przenoszenia do postaci elektronicznej informacji z dokumentów archiwalnych. W pierwszej fazie projektu skupiono się na dokumentach archiwalnych pochodzących z europejskich muzeów mieszczących się w byłych obozach koncentracyjnych, w pierwszym rzędzie włączając do współpracy muzeum obozu Stutthof.

Tworzenie bibliotek cyfrowych dokumentów archiwalnych wymaga rozwiązania nie tylko problemów związanych z ich archiwizacją i udostępnianiem, ale również z automatycznym wydobyciem i przeniesieniem do postaci elektronicznej treści tych dokumentów. Jednak, mimo istnienia wielu zaawansowanych narzędzi rozpoznawania tekstu (*OCR – Optical Character Recognition*) i względnie prostej struktury dokumentów obozowych zachowanych w formie maszynopisów, automatyzacja tego procesu jest zagadnieniem trudnym, rodzącym liczne problemy natury technicznej, naukowej, organizacyjnej oraz prawnej [2,3].

Osiągnięcie odpowiedniej jakości pozyskiwania (rozpoznawania) informacji zależy w bardzo dużym stopniu od zastosowanych metod i narzędzi przetwarzania i rozpoznawania obrazów. W projekcie *Memorial* wykorzystywane jest profesjonalne oprogramowa-

nie OCR *DOKuStar* firmy Document Technologies GmbH. Przeprowadzone badania wykazały, iż jak w większości tego typu programów, jakość rozpoznawania zawartości dokumentu w bardzo dużym stopniu zależy od prawidłowego przeprowadzenia czynności przetwarzania wstępnego, które obejmują:

- pozyskanie dokumentu (skanowanie);
- lokalizację położenia i układu dokumentu na obrazie;
- ewentualną korekcję skosu i/lub zniekształceń geometrycznych;
- usunięcie tła
- lokalizację regionów zawierających rozpoznawany tekst.

W niniejszej pracy przedstawiono wyniki badań związanych z pozyskiwaniem i wstępnym przetwarzaniem dokumentów archiwalnych oraz lokalizacją regionów zawierających tekst na podstawie predefiniowanych szablonów dokumentu.

2. POZYSKIWANIE DOKUMENTÓW ARCHIWALNYCH

Proces pozyskiwania dokumentów archiwalnych rodzi liczne problemy natury technicznej, naukowej, organizacyjnej oraz prawnej. W projekcie *Memorial* przeprowadzono szereg studiów i badań mających na celu określenie właściwego procesu postępowania w przypadku pozyskiwania tych dokumentów.

2.1. Ochrona danych osobowych

Jednym z istotnych aspektów tego procesu jest ochrona danych osobowych zawartych w przetwarzanych dokumentach. Aby skutecznie chronić te dane, w ramach projektu *Memorial* opracowano zestaw falsyfikatów, wykorzystywanych do niektórych badań oraz do prezentacji celów i osiągnięć projektu. Dzięki wykorzystaniu oryginalnego papieru oraz oryginalnej maszyny do pisania, falsyfikaty cechują się wysokim podobieństwem do oryginałów (rys. 1 a).

Przeprowadzone badania wykazały jednak, iż podobieństwo to jest niewystarczające na etapie wstępnego przetwarzania obrazów. Istniejące różnice wynikają z kilku czynników, w tym m.in. z trudności symulowania takich efektów, jak: uszkodzenia mechaniczne papieru, wpływ czynników zewnętrznych (wilgoć, zalania), reakcja tuszu z papierem, przebitki z przylegających arkuszy i wielu innych. Przykładowe uszkodzenia występujące w dokumentach oryginalnych przedstawione zostały na rys. 1. b.

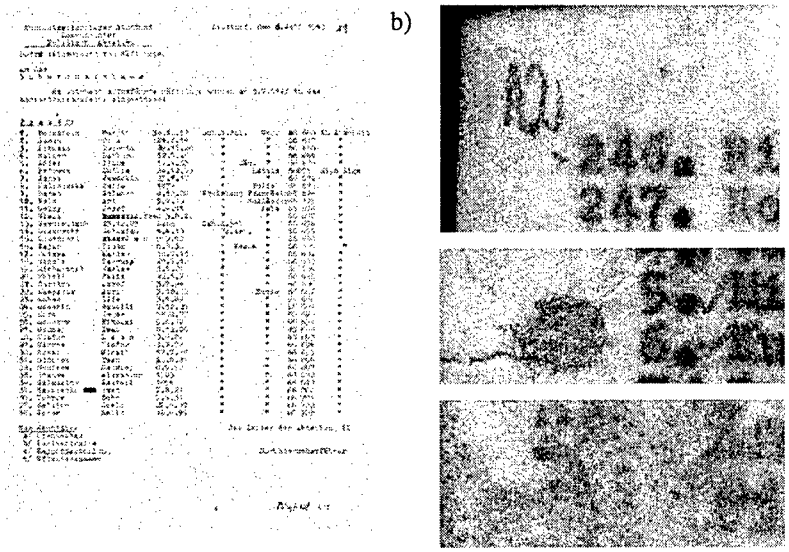
Ze względu na ochronę danych osobowych, przykłady prezentowane w artykule są ograniczone wyłącznie do fragmentów dokumentów oryginalnych nie zawierających takich danych.

2.2. Skanowanie

Podstawowym celem procesu skanowania dokumentów dla potrzeb archiwizacji jest ich wierna konwersja z postaci analogowej (druk) do postaci cyfrowej (obraz). Jednak z punktu widzenia późniejszego procesu rozpoznawania obrazów, tradycyjny proces skanowania może bezpowrotnie gubić istotne informacje. Aby uniknąć tego typu sytuacji w projekcie *Memorial* przeprowadzono dodatkowe badania skanowania obrazów w świetle podczerwonym. Mimo, iż uzyskane w podczerwieni obrazy intensywniej eksponowały pewne własności dokumentów (m.in. tzw. przebitki), to ich ogólna jakość okazała się gorsza od obrazów skanowanych tradycyjnym światłem widzialnym. Teoretycznie istnieje

możliwość wykorzystania informacji z obydwu źródeł, rodzi to jednak dodatkowe problemy związane z dokładnym dopasowaniem uzyskanych obrazów. Dalsze badania w projekcie *Memorial* przeprowadzono już wyłącznie z różnymi typami skanerów tradycyjnych.

a) b)



Rys.1. a) falsyfikat dokumentu obozowego, b) fragmenty dokumentów oryginalnych z przykładowymi uszkodzeniami.

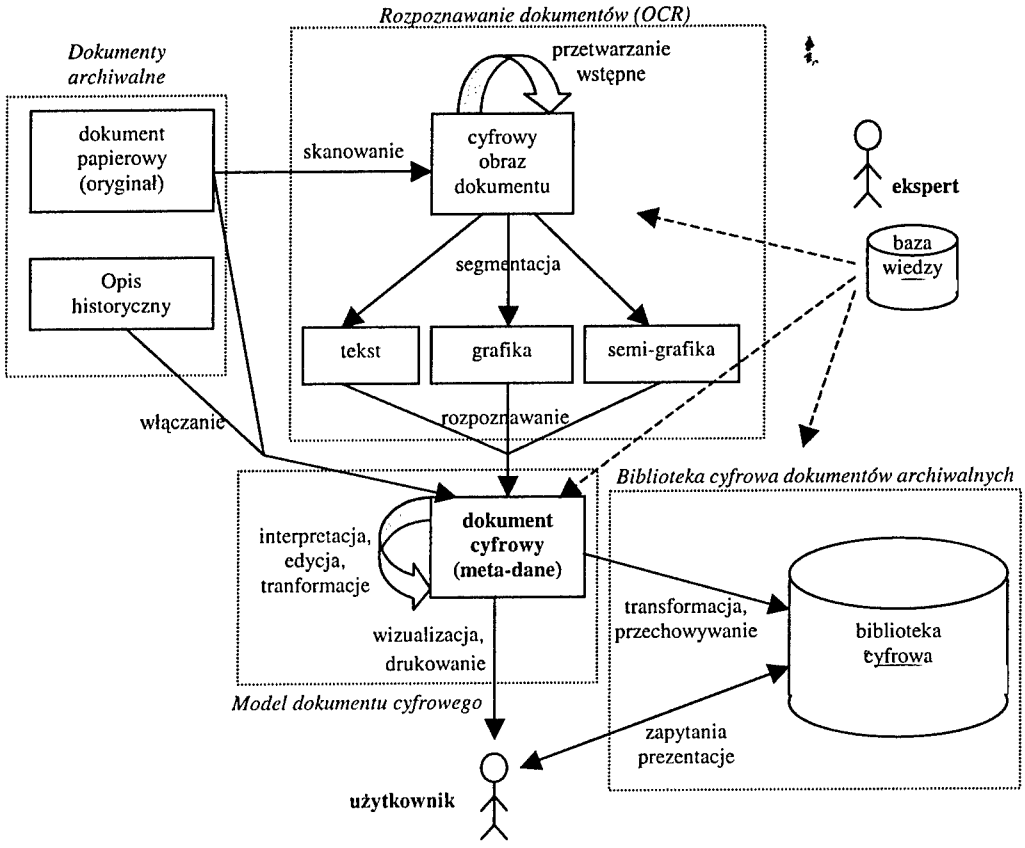
Optymalna rozdzielczość skanowania dokumentów na potrzeby optycznego rozpoznawania obrazów mieści się w większości przypadków w granicach 75-400 DPI [4]. Dolna granica tego przedziału jest określona przez minimalną grubość linii występujących w dokumencie. Linie te po zeskanowaniu powinny mieć grubość ok. 1 do 2 pikseli [5]. Mimo, iż wyznaczony w ten sposób dolny próg dla badanych dokumentów obozowych wynosi ok. 50-75 DPI, w projekcie *Memorial* zastosowano rozdzielczość 300 DPI. Zastosowanie tak wysokiej (w tym przypadku) rozdzielczości wynika z faktu, iż zeskanowane obrazy pełnią równocześnie rolę archiwizacyjną. Obrazy te są prezentowane uprawnionym klientom biblioteki cyfrowej, bez potrzeby udostępniania dokumentów oryginalnych.

Niestety, zastosowanie tak wysokiej rozdzielczości skanowania wybitnie zwiększa zapotrzebowanie na pamięć oraz czas przetwarzania dokumentów, gdyż wielkości te są w przybliżeniu proporcjonalne do kwadratu rozdzielczości obrazu. Przykładowo, obraz pojedynczego dokumentu A4 ma wielkość ok. 26MB, a wykonanie operacji filtracji (oknem 3x3) na takim obrazie trwa[1] ok. 1s. Mimo, iż wszystkie dotychczasowe badania przeprowadzono na obrazach o rozdzielczości 300 DPI, to jest możliwe opracowanie analogicznych algorytmów dla obrazów o zmniejszonej rozdzielczości, co znacząco podniosłoby ich wydajność.

[1] Opisane w artykule badania wydajności algorytmów przeprowadzono na komputerze PC z procesorem Athlon 2.5GHz wyposażonym w 512 MB RAM.

3. PRZETWARZANIE DOKUMENTÓW ARCHIWALNYCH

Proces automatycznego rozpoznawania dokumentu składa się z kilku, charakterystycznych etapów: przetwarzania wstępnego, segmentacji, rozpoznawania oraz interpretacji [2,3]. Etapy te stanowią część cyklu życia dokumentu cyfrowego, który został schematycznie przedstawiony na rys. 2. Skuteczność i efektywność automatycznego procesu rozpoznawania dokumentów cyfrowych decyduje o sukcesie całego projektu tworzenia biblioteki cyfrowej. W przypadku niskiej skuteczności lub wydajności tego procesu, jedyną alternatywą pozostaje ręczne wprowadzanie zawartości dokumentów.



Rys.2. Cykl życia dokumentu cyfrowego

Jak już wspomniano wcześniej, skuteczność rozpoznawania istniejących systemów OCR w dużej mierze zależy od prawidłowego przeprowadzenia czynności wstępnych, a zwłaszcza eliminacji tła oraz segmentacji. Badania przeprowadzone z wykorzystaniem stosowanego w projekcie *Memorial* oprogramowania *DOKuStar*, wykazały jego wysoką wrażliwość na właściwą lokalizację regionów zawierających tekst. W związku z tym, pojawił się problem wyznaczania dokładnego położenia poszczególnych regionów zawierających tekst.

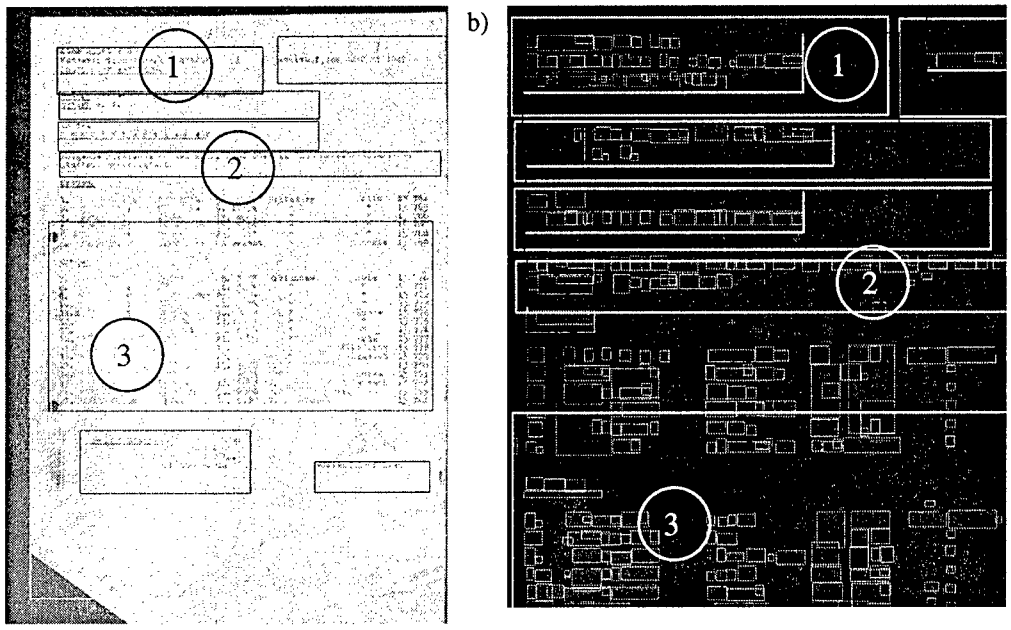
W algorytmach rozpoznawania obrazów dużą rolę odgrywa wiedza eksperta, która znajduje odbicie m.in. w przyjmowanych założeniach, modelach, progach i parametrach

tych algorytmów. W projekcie *Memorial* wiedza eksperta zaszyta jest m.in. w mechanizmie szablonów, które w ogólny sposób opisują strukturę dokumentów danego typu oraz ich wzorcowe położenie na stronie. Szablon zapisany jest w notacji XML i stanowi podstawę do cyfrowego opisu dokumentu. Zawartość poszczególnych pól jest uzupełniana przez algorytmy OCR i ręczną korektę użytkownika. Przykładowy fragment szablonu przedstawiono na rys.3, a położenie wzorcowych regionów na rys. 4 a) (białe prostokąty).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE page SYSTEM "DTD\page.dtd"[]>
<page name="singlepagetransportlist" size_width="2552" size_height="3508">
<content origin_x="0" origin_y="0">
 <region foreground_color="#000000" background_color="#ffffff"
       anchor_top="185" anchor_left="260" stretch_width="1185"
        stretch_height="265" stretch_x_tolerance="0"
        stretch_y_tolerance="0" anchor_x_tolerance="0"
        anchor_y_tolerance="0" skew_tolerance="0" skew="0" optional="False">
   <text font_name="Courier New" charset="ISO-8859-1">
     <composed_text type="">
       <line skip="265" length="0" align="unknown" valign="unknown"></line>
     </composed_text>
   </text>
 </region>
```

Rys.3. Listing fragmentu szablonu XML opisującego wzorcowe położenie segmentów

a) b)



Rys.4. Oryginalny dokument obozowy: a) z zaznaczonymi regionami: wzorcowymi (białe), przesuniętymi (ciemno szare) oraz dopasowanymi (jasno szare), b) wykryte segmenty znaków oraz regiony

3.1. Lokalizacja dokumentu

Niektóre partie dokumentów archiwalnych przetwarzanych w projekcie *Memorial* zostały zeskanowane na tle szarozielonego sukna (rys. 4.a). Pierwszym etapem przetwarzania takich dokumentów jest ustalenie ich położenia w obrębie obrazu. Z uwagi na niezmienny rodzaj (kolor) materiału tła oraz jednakowe warunki oświetlenia (skaner) ustalono charakterystykę koloru tła i zastosowano prosty algorytm próbkujący obraz od brzegu w kierunku środka dokumentu, co zadaną liczbę pikseli w pionie i poziomie. Wystąpienie N kolejnych punktów (N=3..10) o charakterystyce barwnej różnej od sukna oznacza brzeg dokumentu. Mimo, iż przy N = 10 można wyznaczyć obrys dokumentu o dowolnym kształcie (np. ścięte rogi, pofałdowany brzeg itp.), to w praktyce wystarczy zlokalizować najmniejszy prostokąt wpisany w dokument przy N=3 (rys. 4a). Podane podejście, mimo swojej prostoty, jest w pełni skuteczne, a czas jego działania jest niemierzalny.

3.2. Separacja tła

Kolejnym etapem przetwarzania wstępnego maszynopisu jest eliminacja tła, czyli tych fragmentów obrazu maszynopisu, na których nie są umieszczone żadne informacje istotne dla potrzeb późniejszego rozpoznawania. Operacja separacji tła prowadzi do wyodrębnienia znaków maszynowych, natomiast elementy graficzne, takie jak, np. podpisy, czy ręczne adnotacje są w tym wypadku traktowane jako niepożądane zakłócenie i powinny zostać usunięte. Niestety, wspomniana wcześniej niska jakość dokumentów archiwalnych z reguły nie pozwala na stosowanie metod progowania globalnego, a nawet lokalnego [6]. Dodatkowym czynnikiem ograniczającym możliwe do zastosowania metody jest ich wydajność czasowa, co wynika z potrzeby interaktywnego przetwarzania dokumentów w czasie rzeczywistym (*on-line*).

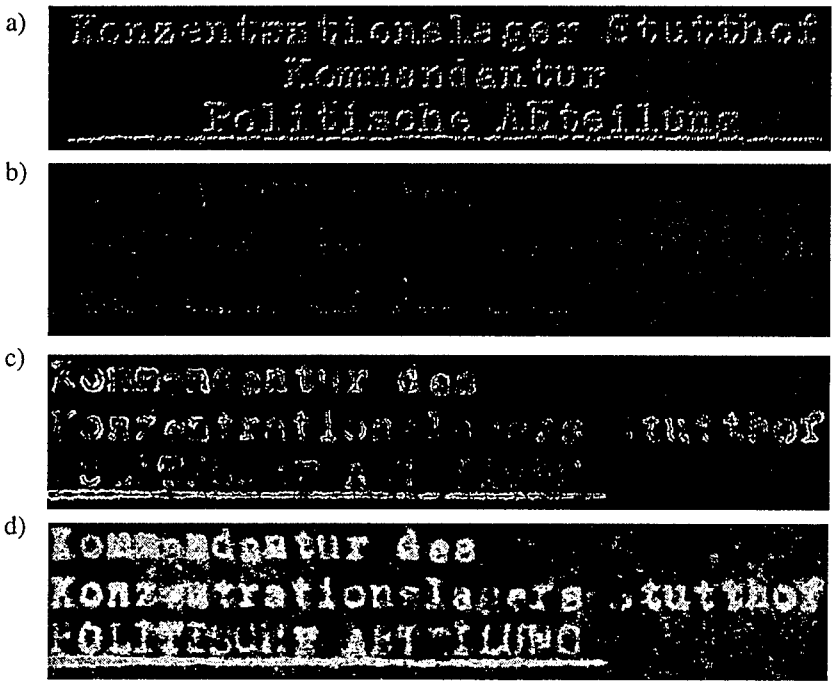
Filtry górno-przepustowe

Zastosowanie filtrów górno-przepustowych do detekcji liter poprzez wykrycie ich krawędzi jest powszechnie stosowanym podejściem. W początkowej fazie prac w projekcie *Memorial* przeprowadzono badania nad segmentacją falsyfikatów z wykorzystaniem zmodyfikowanych filtrów Sobela [4], uzyskując bardzo dobre efekty (rys. 5a). Niestety, próba zastosowania filtrów górno-przepustowych do obrazów oryginalnych nie powiodła się dla masek wielkości 3x3 (rys. 5b). Zwiększenie maski filtru do rozmiaru 5x5 wyraźnie poprawiło jakość detekcji (rys. 5c), wydłużając jednak znacznie czas operacji powyżej założonego progu.

Konieczność odrzucenia filtrów górno-przepustowych na tym etapie badań wynikła ze znacznego rozmycia liter na dokumentach oryginalnych. Filtry te mogą być jednak stosowane lokalnie w celu zwiększenia pewności odpowiedzi innych metod detekcji znaków.

Progowanie obrazów kolorowych

Podczas stosowania systemów OCR do rozpoznawania obrazów dobrej jakości wystarczające są obrazy monochromatyczne. Jednakże w dokumentach archiwalnych znaczna część informacji jest zapisana również w składowych chrominancji. Odrzucenie tej części informacji może w takim przypadku prowadzić do nieprawidłowej separacji tła, a w konsekwencji do błędnego rozpoznania zawartości dokumentu.



Rys.5. Wykrywanie znaków (separacja tła) przy użyciu: filtrów Sobela (a – dla falsyfikatu, b – dla oryginału z maską 3x3, c – dla oryginału z maską 5x5) oraz d – algorytmu dla obrazów kolorowych

Badania przeprowadzone w ramach projektu *Memorial* pozwoliły na określenie algorytmu konwersji obrazów dokumentów z postaci kolorowej (RGB) na monochromatyczną w sposób eliminujący znakomitą większość obszaru tła (rys.5d). Metoda wyznaczania punktów tła jest złożeniem 3 metod bazujących na progach ι wyznaczonych statystycznie. Istotne jest, że wybór progów nie jest krytyczny, tzn. zapewniają one bezpieczny margines błędu dla różnych dokumentów. Wykorzystywanymi w algorytmie metodami są:

- odrzucenie punktów należących do tła dokumentu (zob. 3.1). Punkty takie pojawiają się również wewnątrz wyznaczonych marginesów położenia dokumentu w obrazie graficznym w sytuacji, gdy dokument jest zniszczony fizycznie i sukno jest widoczne (dziury po dziurkaczu, ścięte narożniki lub poszarpane brzegi) lub tylko prześwituje przez zbyt cienki papier.
- progowanie jasności J z dwoma progami ι_{Jmin} (=50) oraz ι_{Jmax} (=180) wyznaczające punkty, które na pewno nie są literami; punkty zbyt ciemne ($J<\iota_{Jmin}$) występują w znakach bardzo rzadko z uwagi na blaknięcie tuszu maszynowego; punkty zbyt jasne ($J>\iota_{Jmax}$) należą do tła liter.
- filtrowanie punktów sąsiadujących (z prawej strony) co najmniej z jednym punktem tła dokumentu (sukna). Celem tej operacji jest wykrycie większych, spójnych obszarów punktów tła dokumentu. Zastosowanie filtru logicznego z maską o wielkości 3x1 znacznie przyspieszyło działanie algorytmu, choć lepsze efekty daje maska o rozmiarze 3x3.

- odrzucanie punktów „kolorowych" tj. takich, w których dominuje jedna ze składowych koloru RGB (*Red*, *Green*, *Blue*). Wychodząc z założenia, że oryginalne znaki maszynowe były w kolorze czarnym można przyjąć, że z upływem czasu blaknięcie postępuje w zbliżonym stopniu dla poszczególnych składowych koloru. W algorytmie odrzucane są piksele, dla których dowolna ze składowych koloru przewyższa którąkolwiek z pozostałych o zadany próg ι_{chrom} (=30). Przyjęcie tak dużej wartości ι_{chrom}, wspólnej dla wszystkich składowych koloru daje bardzo dobre efekty, przy zachowaniu szerokiego marginesu odporności algorytmu. Badania wykazały, że zmniejszając próg ι_{chrom}, indywidualnie dla poszczególnych składowych RGB można uzyskać jeszcze lepszą separację tła, za wyjątkiem miejsc przebarwionych.

3.3. Detekcja segmentów

W celu określenia rzeczywistych granic regionów tekstowych, należy określić położenie segmentów poszczególnych znaków wchodzących w ich skład. Ponieważ celem nie jest tu rozpoznawanie znaków, a jedynie lokalizacja ich segmentów, można wykorzystać metody uwydatniania segmentów, które nie zachowują ich kształtu. Algorytm, opracowany na potrzeby projektu *Memorial*, składa się z dwóch etapów:

- usunięcia szumu filtrem logicznym o rozmiarze 7x7. Pozostawiane są wyłącznie punkty posiadające co najmniej 5 „sąsiadów" wewnątrz scentrowanego na nich okna.
- morfologicznego rozmycia obrazu (*dilation*) prostokątnym elementem strukturalnym o rozmiarze 9x3 w celu połączenia oderwanych segmentów znaków

Efektem działania algorytmu jest lista spójnych segmentów zawierających pojedyncze lub sklejone znaki maszynopisu.

3.4. Dopasowanie regionów

Zadaniem algorytmu dopasowania regionu jest znalezienie jego rzeczywistego położenia w dokumencie na podstawie zadanego położenia wzorcowego podanego w szablonie. Zadanie to wykonywane jest w dwóch wersjach:

1. Znalezienie rzeczywistego położenia lewego górnego rogu wzorca, przy zachowaniu jego oryginalnego rozmiaru. Znalezione w ten sposób regiony zaznaczono na rys. 4 ciemnoszarymi (niebieskimi) prostokątami.
2. Znalezienie najmniejszego prostokąta opisującego znaki w regionie. Znalezione w ten sposób regiony zaznaczono na rys. 4 jasnoszarymi (żółtymi) prostokątami.

Zastosowany do realizacji tych celów algorytm znajduje wszystkie segmenty o odpowiedniej wielkości, które leżą wewnątrz segmentu wzorcowego lub zachodzą na niego przynajmniej w $\iota_{intersect}$ =40%. Minimalne wymiary segmentów zależą od wielkości czcionek i dla dokumentów obozowych zostały określone na $\iota_{minsegx} = \iota_{minsegy} = 15$ pikseli.

Zadania stawiane przed algorytmem wymagają, by region wzorcowy obejmował swym zasięgiem wszystkie znaki do niego przynależące. Przykłady regionów spełniających ten wymóg oznaczono cyframi 1 i 2 na rys. 4. Jeżeli powyższy warunek nie jest spełniony, algorytm rozszerza wymiary regionu tak, by obejmowały najbliższe znaki, brak mu jednak podstaw, by kontynuować rozszerzanie na kolejne wiersze lub kolumny znaków. Przykład takiej sytuacji oznaczono na rys. 4 cyfrą 3. Próby rozszerzenia regionów o kolejne wiersze wymagałby znajomości struktury całego dokumentu oraz analizy semantycznej sąsiednich

segmentów (np. czy tworzą kolejny wiersz / kolumnę w tabeli), co wykracza poza przyjęte założenia. Tym niemniej, dalsze prace w ramach projektu *Memorial*, będą podążały również w tym kierunku.

4. ZAKOŃCZENIE

W artykule przedstawiono problemy związane przetwarzaniem wstępnym i segmentacją dokumentów archiwalnych złej jakości. Przedstawiono opracowane algorytmy lokalizacji dokumentu, separacji tła, wyszukiwania segmentów oraz dopasowania regionów. Działanie algorytmów zostało zweryfikowane dla ponad 50 dokumentów oryginalnych przynależących do 4 klas, różniących się układem regionów. W obrębie każdej klasy poszczególne dokumenty różniły się położeniem oraz wielkością segmentów. Uzyskane wyniki potwierdziły bardzo dobrą skuteczność algorytmów separacji tła dokumentów, potwierdzając tezę o konieczności wykorzystywania pełnej informacji o obrazie dla dokumentów archiwalnych, w tym również informacji o jego chrominancji.

Wyniki uzyskane przy dopasowaniu regionów były również bardzo dobre, pod warunkiem zachodzenia na siebie prostokątów opisujących: regiony wzorcowe (z szablonu) oraz rzeczywiste położenie regionów. W przeciwnym wypadku znajdowane są dopasowania fragmentaryczne lub brak jest dopasowań. Niemniej jednak, duża szybkość opracowanych algorytmów pozwala na interaktywną pracę z rozpoznawanymi dokumentami, a tym samym na ewentualną ręczną korektę źle określonych regionów. Zadanie to ułatwiają miary określające jakość takiego dopasowania [2,3].

Uzyskane wyniki skłaniają do poszukiwania bardziej zaawansowanych metod wysokiego poziomu uwzględniających wiedzę o globalnej strukturze regionów w dokumencie. Jedną z możliwych do zastosowania metod jest poszukiwanie dopasowania grafów. Badania w tym kierunku będą rozwijane w dalszych pracach nad projektem *Memorial*.

BIBLIOGRAFIA

[1] Wiszniewski B.: *The Virtual Memorial Project*. http://docmaster.eti.pg.gda.pl.

[2] J.Lebiedź, A.Podgórski, M.Szwoch: *Pomiar jakości procesu automatycznego rozpoznawania maszynopisów*, Zeszyty Naukowe WETI PG nr 2 (Mat. Konf. TI 2003), Gdańsk 2003.

[3] J.Lebiedź, A.Podgórski, M.Szwoch: *Quality Evaluation of Computer Aided Information Retrieval from Machine Typed Paper Documents*, Third Confrence on Recogntion Systems KOSYR'2003 (proceedings) Technical Uniwersity of Wrocław, Wrocław 2003.

[4] Malina W., Ablameyko S. Pawlak W.: *Podstawy cyfrowego przetwarzania obrazów*, monografia, Warszawa, Akad. Ofic. Wyd. EXIT 2002.

[5] Szwoch M: *Rozpoznawanie drukowanej notacji muzycznej z wykorzystaniem bezkontekstowych gramatyk atrybutowych*, praca doktorska, WETI PG, Gdańsk 2002.

[6] Sahoo P.K. *et al*: *A Survey of Thresholding Techniques*, CVGIP 41, 1988.

PREPROCESSING AND SEGMENTATION OF BAD QUALITY MACHINE TYPED PAPER DOCUMENTS

Summary

The goal of the international project Memorial is automatic retrieval from machine typed paper documents. In this paper the problem of pre-processing and segmentation of scanned archival documents is considered. The goal of these processes is to exactly determine text regions in the document for further OCR processing.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Wojciech Waloszek

Katedra Inżynierii Oprogramowania, Politechnika Gdańska, WETI

WYWODZENIE SCHEMATU Z DANYCH O NIEZNANEJ STRUKTURZE ZA POMOCĄ ALGORYTMU COBWEB S+T

Streszczenie

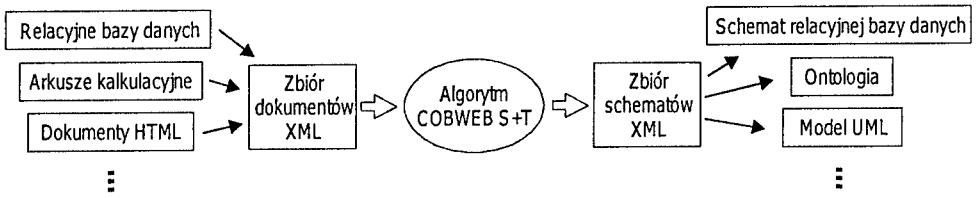
W niniejszym artykule przedstawiono algorytm COBWEB S+T służący do wywodzenia schematów XML ze zbioru dokumentów XML o nieznanej strukturze. Algorytm ten wykorzystuje model danych semistrukturalnych oraz algorytm grupowania pojęciowego COBWEB. Na potrzeby algorytmu COBWEB S+T opracowano funkcje oceny wzajemnego podobieństwa typów i schematów.
Algorytm COBWEB S+T został zaimplementowany wraz ze środowiskiem służącym do testowania algorytmu. W artykule przedstawione są wyniki przeprowadzonych testów wywodzenia schematów: z pojedynczych dokumentów XML, ze zbiorów dokumentów XML zbudowanych według jednego schematu, a także ze zbiorów dokumentów XML zbudowanych według różnych schematów.

1. WSTĘP

Aby odczytać i efektywnie przetwarzać dane, potrzebna jest znajomość zasad, według których dane te są zbudowane, czyli schematu danych. Artykuł prezentuje algorytm COBWEB S+T, który służy do wywodzenia zbioru schematów XML z kolekcji dokumentów XML. Algorytm COBWEB S+T koncentruje się na wywodzeniu informacji o logicznej strukturze zgromadzonych danych, konstruując opisujący przetwarzane dokumenty schemat XML, jeśli dokumenty mają jednolitą strukturę, bądź zbiór schematów XML, jeśli struktury pewnych grup dokumentów wyraźnie się różnią.

Algorytm COBWEB S+T można zastosować jako element bardziej złożonego systemu przetwarzania danych, tak jak pokazuje to rysunek 1. Za pomocą ogólnodostępnych konwerterów różnorodne dane mogą zostać przetworzone do postaci dokumentów XML. Wywiedziony zbiór schematów może być potraktowany zarówno jako finalny produkt przetwarzania danych (kiedy celem jest np. ujednolicenie formatu danych dotychczas przechowywanych w różnych postaciach i opracowanie zbioru schematów opisujących te dane), jak i jako przedmiot dalszej analizy i dalszego przetwarzania.

W algorytmie COBWEB S+T wykorzystano specjalnie zaadaptowaną wersję algorytmu COBWEB, znanego z systemów uczących się ([1], [2]). Zmiany dokonane w oryginalnym algorytmie COBWEB są dość poważne. Obejmują zdefiniowanie funkcji mierzących podobieństwo schematów i typów XML oraz zmianę opisu drzewa kategorii obiektów generowanego przez COBWEB.



Rys.1. Potencjalne zastosowania algorytmu COBWEB S+T.

Niektóre mechanizmy użyte w algorytmie COBWEB S+T pochodzą z prac nad semistrukturalnym modelem danych ([3], [4], [5]). Prace te, obejmujące analizę danych o nieznanej strukturze, przyczyniły się znacznie do rozwoju technik związanych z wywodzeniem schematów z kolekcji dokumentów XML ([6], [7], [8]).

2. ALGORYTM COBWEB S+T

Algorytm, w najogólniejszej postaci, przedstawiony jest na rysunku 2.

```
Wejście:   D: Zbiór dokumentów XML.
Wyjście:   S: Zbiór schematów.
  1: for each d in D
  2:   Wywiedź informacje o schemacie z dokumentu d.
  3:   Dołącz wywiedzione informacje do zbioru I.
  4: next
  5: Wygeneruj optymalny zbiór schematów S na podstawie I.
```

Rys.2. Ogólna postać algorytmu COBWEB S+T.

Zadaniem algorytmu jest wywiedzenie optymalnego zbioru schematów, przy czym kryteriami optymalizacji są:

- zwięzłość – im mniej schematów, tym lepszy zbiór,
- precyzja – im lepiej dopasowane do schematów są dokumenty im przyporządkowane, tym lepszy zbiór.

2.1. Wywodzenie schematu z pojedynczego dokumentu XML

Wywodzenie schematu z pojedynczego dokumentu XML jest szczególnym przypadkiem postawionego wyżej zasadniczego problemu. Wywodzenie takie odbywa się w kilku etapach (patrz rysunek 3), które opisane są w kolejnych punktach.

```
Wejście:   d: Dokument XML.
Wyjście:   s: Schemat XML.
  1: Zidentyfikuj największy zbiór opisów typów T.
  2: Pogrupuj wywiedzione typy za pomocą drzewa COBWEB T.
  3: Dokonaj redukcji zbioru T; wynik umieść w zbiorze R.
  4: Wygeneruj s na podstawie zbioru R.
```

Rys.3. Algorytm wywodzenia schematu XML z pojedynczego dokumentu XML.

2.1.1. Przetwarzanie wstępne dokumentu

Przetwarzanie wstępne ma na celu odnalezienie pierwszego, maksymalnego zbioru typów. Tak wydobyty zbiór typów podlega dalszemu przetwarzaniu. Do przetwarzania wstępnego wykorzystano algorytm tworzenia przewodników danych (służący do analizy danych semistrukturalnych, [3]). W dokumencie XML identyfikowane są najpierw wszystkie ścieżki. Następnie wyznaczane są zbiory elementów, do których prowadzą poszczególne ścieżki. Na podstawie zawartości elementów należących do tak wyznaczonych zbiorów budowany jest opis typu zawierający:

- zbiór opisów atrybutów,
- zbiór opisów elementów podrzędnych,
- opis kolejności elementów podrzędnych,
- wskaźnik mówiący, czy element może zawierać tekst.

Opis atrybutu zawiera jego nazwę i *krotność*. Atrybuty opcjonalne mają krotność `"?"`, a wymagane `"1"`. Opis elementu podrzędnego składa się z nazwy, krotności określającej, czy element jest wymagany i powtarzalny, oraz z odnośnika do typu elementu podrzędnego (opisu typu przypisanego ścieżce wydłużonej o nazwę elementu podrzędnego). Opis kolejności elementów podrzędnych ma postać grafu skierowanego, którego wierzchołkom są przypisane elementy podrzędne, zaś istnienie łuku między elementami A i B oznacza, że element podrzędny A poprzedza element podrzędny B.

Niech zbiór E zawiera elementy, do których prowadzi pewna ścieżka. Na podstawie tego zbioru konstruowany jest opis typu. Wszystkie atrybuty i elementy podrzędne, zawarte w elementach z E, są rejestrowane w opisie typu. Przypisywane są im najbardziej ograniczające krotności, opisy kolejności i zawartości tekstowej, jednak takie, że opis typu wciąż poprawnie opisuje wszystkie elementy z E.

2.1.2. Grupowanie typów

Wstępne przetwarzanie dokumentu wystarcza do utworzenia schematu na podstawie dokumentu XML. Jednak mechanizm ten tworzy zbyt wiele typów, ignorując fakt, że różne ścieżki mogą prowadzić do bardzo podobnych elementów. Tworzy on więc schematy o wysokiej precyzji i bardzo niskiej zwięzłości. Wskazane jest dalsze przetworzenie zbioru typów, tak by zidentyfikować i połączyć podobne typy.

Algorytm użyty w tym celu to COBWEB ([1], [2]). Grupuje on podane na wejściu obiekty, może być więc użyty również do wskazania grup podobnych typów.

COBWEB generuje hierarchię (drzewo) kategorii obiektów, poprzez przeszukiwanie przestrzeni możliwych grupowań. Każdy obiekt umieszczany jest w drzewie przez wykonywanie kilku prostych operacji (szczegóły w [1]). Decyzja, którą operację wykonać, podejmowana jest na podstawie funkcji oceny jakości grupowania f:

$$f(h) = ZWZ(|C_h|)\sum_{d\in C_h}\left[WAGA(d)\cdot(PREC(d) - PREC(d_0))\right] \tag{1}$$

gdzie C_h to zbiór podkategorii przy grupowaniu h, d_0 to kategoria bieżąca, *ZWZ* to współczynnik zwięzłości (odwrotność liczby podkategorii), *WAGA* to waga podkategorii d (stosunek liczby obiektów w d do liczby obiektów w d_0), a *PREC* to współczynnik precyzji, mierzący wzajemne podobieństwo obiektów w ramach podkategorii.

Widać więc, że COBWEB uwzględnia kryteria precyzji i zwięzłości; ponadto definiuje on dla tych kryteriów miary. Miary te zakładają jednak, że grupowane obiekty opisane są

za pomocą krotek. Funkcja *PREC* zdefiniowana jest w terminach estymatorów prawdopodobieństwa tak, że jej wartość jest tym wyższa, im więcej przykładów o takich samych wartościach poszczególnych atrybutów znajduje się w danej podkategorii.

Niestety, ten sposób opisu obiektów uniemożliwia bezpośrednie użycie algorytmu COBWEB do grupowania typów, jako że opisy typów trudno zapisać w postaci krotki (szczegółowo problem ten opisany jest w [9]). Aby rozwiązać ten problem przedefiniowano sposób opisu kategorii obiektów i miarę podobieństwa obiektów.

Opis kategorii typów generowany jest na podstawie opisów typów znajdujących się w danej podkategorii. Podobnie jak opis typu, opis kategorii typów zawiera:

- zbiór opisów atrybutów,
- zbiór opisów elementów podrzędnych,
- opis kolejności elementów podrzędnych,
- wskaźnik mówiący czy dane elementy mają zawartość tekstową,
- liczbę typów zawartych w kategorii.

Atrybuty i elementy opisywane są podobnie, jak w przypadku opisu typu, jednak krotności atrybutów i elementów podrzędnych zostały zastąpione zbiorami odpowiednich krotności, przy czym została dopuszczona dodatkowa, sztuczna krotność "0", oznaczająca, że w jednym z typów w ramach kategorii atrybut lub element nie występuje. Odnośnik do typu elementu podrzędnego zastąpiono odnośnikiem do najmniejszej kategorii zawierającej wszystkie wymagane typy danego elementu podrzędnego. Opis kolejności elementów w ramach kategorii to graf skierowany składający się z sumy wierzchołków i łuków z opisów wszystkich typów zawartych w kategorii. Wskaźnik zawartości tekstowej jest zbiorem wartości takiej samej właściwości w ramach opisów zawartych w kategorii typów. Liczba typów w kategorii jest potrzebna do obliczenia wartości współczynników *ZWZ* i *PREC*.

Na podstawie takiego opisu kategorii typów można przeprowadzić ocenę wzajemnego podobieństwa typów zawartych w kategorii. Czynniki mające wpływ na tę ocenę to precyzja opisu atrybutów oraz precyzja opisu elementów podrzędnych. Czynnikom tym przypisano liczby z zakresu <0,1>. Precyzja opisu atrybutu jest mierzona na podstawie zbioru jego krotności. Każdemu zbiorowi krotności przypisano heurystycznie pewną liczbę. Ocena precyzji opisu elementu podrzędnego jest iloczynem dwóch czynników. Pierwszy to ocena zbioru krotności, liczona analogicznie jak dla atrybutów. Drugi czynnik to ocena precyzji typu przypisanego danemu elementowi, liczona jako ocena precyzji kategorii typów, do której prowadzi odpowiedni odnośnik. Ostatecznie, ocena precyzji kategorii typów wyznaczana jest jako średnia ocen precyzji opisów poszczególnych atrybutów i elementów podrzędnych:

$$PREC_T(d) = \frac{\sum_{i=1}^{m} f_A(A_i) + \sum_{i=1}^{n} f_E(E_i)}{m+n}, \text{ lub } 0 \text{ gdy } m = n = 0 \qquad (2)$$

gdzie d to opis kategorii typów, E_i to opis i-tego elementu podrzędnego, A_i to opis i-tego atrybutu, f_A i f_E to funkcje oceny precyzji odpowiednio opisu atrybutu i opisu elementu, m to liczba atrybutów w d, a n to liczba elementów podrzędnych w d. $PREC_T$ jest funkcją rekurencyjną, gdyż f_E zawiera $PREC_T$. Wartość $PREC_T$ obliczamy od liści (dla liści $n = 0$).

Funkcja oceny precyzji kategorii typów może być łatwo rozszerzana o kolejne czynniki, związane np. z kolejnością elementów podrzędnych lub zawartością tekstową.

2.1.3. Redukcja typów

Możliwość wyznaczenia precyzji kategorii typów ułatwia wybór kategorii do redukcji. Kategoria jest *redukowalna*, gdy jej precyzja przekracza pewien zdefiniowany próg precyzji i wszystkie kategorie, do których prowadzą odnośniki, są redukowalne.

Największe redukowalne kategorie są wybierane do redukcji i używane do utworzenia ostatecznego zbioru typów. Opisy atrybutów, elementów podrzędnych, kolejności i zawartości tekstowej są zamieniane na opis pojedynczego typu. Odnośniki między typami (określające typy elementów podrzędnych) są generowane na podstawie odnośników między kategoriami.

Wynikowy zbiór typów spełnia kryteria precyzji i zwięzłości. Algorytm próbuje wyważyć oba kryteria i znaleźć optimum, generując minimalną liczbę typów przy zadanym progu precyzji.

2.1.4. Generowanie schematu XML na podstawie zbioru opisów typów

Ostatni etap wywodzenia schematu z pojedynczego dokumentu polega na generacji schematu XML z określonego wcześniej zbioru opisów typów. Każdy opis typu zostaje zamieniony na definicję typu XML. Atrybut `mixed` definicji typu jest ustawiany zgodnie ze wskaźnikiem zawartości tekstowej. Opisy atrybutów zamieniane są na deklaracje atrybutów. Krotność określa atrybut jako wymagany (krotność `"1"`) lub opcjonalny (krotność `"?"`).

Model zawartości typu XML definiowany jest na podstawie opisu kolejności. Grupy elementów połączonych w tym opisie pełnymi podgrafami stanowią *grupy równorzędne*. Grupy te są porządkowane w jak najdłuższą sekwencję nienaruszającą opisu kolejności. Ta sekwencja staje się podstawą modelu zawartości. Grupy równorzędne zawierające tylko jeden wierzchołek są zamieniane na deklarację elementu podrzędnego w sekwencji (atrybuty `minOccurs` i `maxOccurs` ustawiane są według krotności). Definicje elementów z większych grup równorzędnych umieszczane są wewnątrz grupy `choice` o nieograniczonej liczbie wystąpień.

Schemat XML musi też zawierać deklarację globalną głównego elementu dokumentu. Aby przypisać właściwy typ głównemu elementowi, algorytm musi pamiętać, do której z kategorii typów został zaliczony typ głównego elementu.

2.2. Grupowanie schematów

Algorytm z rys. 3 może być użyty do wywiedzenia kolekcji schematów XML z kolekcji dokumentów XML; otrzymamy wówczas jeden schemat dla każdego dokumentu. Taki zbiór schematów, choć bardzo precyzyjny, będzie jednak mało zwięzły. Tak jak w przypadku typów, rozwiązaniem jest grupowanie schematów za pomocą zaadaptowanej wersji drzewa COBWEB, nazwanej COBWEB S.

Opis kategorii schematów zawiera pary <ścieżka, opis typu>, wygenerowane na drodze przetwarzania wstępnego, przed redukcją, i liczbę typów po redukcji (*rozmiar schematu*). Opis schematu generowany jest za pomocą już opisanych mechanizmów służących do wywiedzenia pojedynczego schematu XML z dokumentu (patrz rys. 4).

```
Wejście:   d: Dokument XML.
Wyjście:   s: Opis schematu.
  1: Zidentyfikuj wstępny zbiór typów T;
     zapamiętaj pary <ścieżka, opis typu> w s.
  2: Pogrupuj wywiedzione typy za pomocą drzewa COBWEB T.
  3: Dokonaj redukcji zbioru typów T; umieść wynik w zbiorze R.
  4: Wyznacz liczbę typów w R i zapamiętaj ją w s.
```

Rys. 4. Tworzenie opisu schematu w algorytmie COBWEB S+T.

Opisy schematów są grupowane w kategorie schematów zawierające dane opisujące połączenie kilku schematów w jeden. Informacja o typach przechowywana jest w postaci par <ścieżka, opis kategorii typów>; ponadto kategoria zawiera: łączną liczbę typów po redukcji w schematach z danej kategorii (każdy schemat traktowany oddzielnie) i liczbę typów po redukcji w całej kategorii (tj. obejmującej zbiór wszystkich typów ze wszystkich schematów w danej kategorii).

Kiedy nowy schemat jest dodawany do kategorii, opisy typów z nowego schematu są dodawane do kategorii typów przypisanych odpowiednim ścieżkom. Jeśli kategoria schematów nie zawiera pewnej ścieżki, ścieżka jest dodawana do kategorii w parze z nową kategorią typów zawierającą jedynie typ z nowego schematu. W drugim kroku obliczana jest łączna liczba typów po redukcji. Krok trzeci obejmuje wyznaczenie liczby typów po redukcji w całej kategorii. Najpierw z kategorii typów generowane są opisy typów, które następnie przetwarzane są za pomocą algorytmu COBWEB T, tak jak opisano to w punktach 2.1.2 i 2.1.3. Liczba typów po redukcji zapamiętywana jest w opisie kategorii. Tę liczbę można też określić jako *rozmiar połączonego schematu.*

Na ocenę precyzji kategorii schematów składają się trzy czynniki. Pierwszy jest oceną precyzji kategorii typów odpowiadających poszczególnym ścieżkom, oznaczaną $f_P(d)$ i obliczaną jako średnia ocen precyzji kategorii typów zawierających więcej niż jeden typ:

$$f_P(d) = \frac{1}{|T2|} \sum_{t \in T2} PREC_T(t) \tag{3}$$

gdzie $T2$ to zbiór kategorii typów zawierających co najmniej dwa typy, a $PREC_T$ to funkcja oceny precyzji kategorii typów zdefiniowana wzorem (2).

Drugi z czynników związany jest z zyskiem na liczbie typów, który udało się osiągnąć dzięki połączeniu kilku schematów w jeden schemat. Zysk ten jest normalizowany względem liczby schematów w kategorii oraz średniego rozmiaru schematu w kategorii. Współczynnik zysku f_Z obliczamy następująco:

$$f_Z(n, R, r) = \frac{R - r}{(R/n) \cdot (n-1)} \tag{4}$$

gdzie n jest liczbą schematów w d, R to łączna liczba typów po redukcji, a r to rozmiar połączonego schematu.

Ostatni czynnik to iloraz średniego rozmiaru schematu w kategorii przez rozmiar połączonego schematu; hamuje on tendencję tworzenia dużych połączonych schematów:

$$f_R(n, R, r) = \frac{(R/n)}{r} \tag{5}$$

Ocena precyzji kategorii schematów d (oznaczana $PREC_S(d)$) jest iloczynem tych trzech czynników.

W rezultacie działania algorytmu COBWEB S powstaje drzewo kategorii schematów. Dla każdej kategorii obliczana jest ocena precyzji. Następnie wybierane są największe kategorie schematów, których ocena przekracza *próg prezycji*, i na ich podstawie generowany jest ostateczny zbiór schematów XML.

3. WYNIKI EKSPERYMENTÓW

Aby sprawdzić skuteczność algorytmu, zdecydowano się na jego implementację wraz ze środowiskiem pozwalającym na jego testowanie. Jako pierwsze przeprowadzono testy na pojedynczych dokumentach XML. Algorytm okazał się zdolny do utworzenia poprawnego schematu XML oraz do zidentyfikowania typowych struktur spotykanych w dokumentach XML m.in. identycznych poddrzew i struktur rekurencyjnych.

Następnie przeprowadzono testy na zbiorach dokumentów. Do testów użyto dwóch zbiorów dokumentów. Pierwszy zbiór składał się z 498 dokumentów pozyskanych z NASA [10]. Drugi zbiór pochodził z [11] i zawierał 37 dramatów Williama Szekspira. Pierwsze testy polegały na przetwarzaniu jednorodnego zbioru dokumentów (tj. dokumentów zbudowane według tego samego DTD). Wyniki testów pozwalają stwierdzić, że algorytm zachowywał się poprawnie. Rozpoznawał wszystkie typy zdefiniowane w DTD, większość wygenerowanych definicji było identycznych z oryginalnymi. Te typy, które w DTD były podobne, zgodnie z oczekiwaniami, łączył, tworząc pojedynczą definicję typów. Występował również efekt odwrotny, przypisanie elementom o jednej nazwie różnych typów w zależności od kontekstu. Efekt ten występował rzadziej, ale pozwalał na wyciągnięcie ciekawych wniosków dotyczących budowy dokumentów.

Celem kolejnych testów było sprawdzenie zdolności algorytmu do rozróżnienia dokumentów zbudowanych według różnych definicji. Tym razem przetwarzano dokumenty XML z obu zbiorów. W pierwszym eksperymencie dokumenty z obu zbiorów podawano na wejście naprzemiennie. Algorytm rozróżnił typy dokumentów XML, tworząc dwa wynikowe schematy XML: jeden dla sztuk Williama Szekspira, drugi dla dokumentów NASA. Otrzymane schematy nie różniły się od schematów otrzymanych w wyniku przetworzenia jednorodnych zbiorów. W następnym teście zmieniono kolejność podawania dokumentów i przetworzono kolejno 20 dokumentów NASA i 20 dramatów Szekspira. Rezultaty były niezadowalające, jako że otrzymano 10 schematów dla dokumentów NASA i 1 dla dramatów Szekspira. Dzieje się tak dlatego, że jedynym dostępnym oryginalnemu algorytmowi COBWEB działaniem, które zmniejsza liczbę kategorii, jest scalenie dwóch najlepszych kategorii dla nowego obiektu. W wyżej opisanej sytuacji dwie najlepsze kategorie dla dramatu musiałyby być kategoriami zawierającymi dokumenty NASA i dramaty, a po złączeniu takich kategorii ocena precyzji byłaby bardzo niska. Kolejne przetwarzanie dokumentów NASA poprawia sytuację i liczba kategorii ulega zmniejszeniu.

Podsumowując, algorytm COBWEB S+T okazał się zdolny do rozróżnienia dokumentów XML o różnej strukturze i utworzył dla nich osobne kategorie schematów. Jednak rezultaty okazały się uzależnione od kolejności przetwarzania dokumentów; najlepsze wyniki otrzymano, gdy dokumenty o różnych strukturach przetwarzano naprzemiennie. Rozwiązaniem tej sytuacji może być wprowadzenie nowego działania dla drzewa COBWEB, polegającego na tworzeniu nowej kategorii głównej, której podkategoriami byłyby dotychczasowa kategoria główna i kategoria zawierająca nowy obiekt. Działanie to jest obecnie przedmiotem badań.

4. PODSUMOWANIE I KIERUNKI DALSZYCH PRAC

Zaprezentowany algorytm łączy techniki grupowania pojęciowego i przetwarzania danych semistrukturalnych, wprowadzając nową technikę wywodzenia schematów z dokumentów XML. Algorytm działa na przecięciu ważnych dziedzin informatyki: sztucznej inteligencji, baz danych i zarządzania wiedzą. Kryteria którymi się kieruje, bazują na definiowalnych funkcjach. Funkcje te mogą być łatwo rozszerzone tak, aby uwzględnić indywidualne potrzeby, np. wszystkie cechy nowego standardu opisu dokumentów, jakim jest XML Schema ([12], [13]).

Dodatkową zaletą algorytmu COBWEB S+T jest fakt, że działa on w trybie inkrementacyjnym (*on-line*). Przy obecnym rozwoju Internetu jest to cecha ważna, gdyż pozwala wykorzystać algorytm w aplikacjach pobierających dane z sieci w sposób ciągły. COBWEB S+T próbuje wywieść optymalny zbiór schematów z zebranych danych, korygując wyniki w miarę napływania nowych dokumentów.

Poza wprowadzeniem nowego działania dla drzewa COBWEB, kierunki rozwoju algorytmu obejmują badanie wpływu parametrów na jego zachowanie, technik automatycznego doboru parametrów w zależności od rodzaju danych, a także wszelkie badania zachowania algorytmu w praktyce, w rzeczywistych zastosowaniach.

BIBLIOGRAFIA

[1] P. Cichosz, *Systemy uczące się*, WNT, Warszawa 2000.
[2] I. H. Witten, E. Frank, *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.
[3] S. Abiteboul, P. Buneman, D. Suciu, *Dane w sieci WWW*, tl. P. Brągoszewski, P. Pęzik, S. Dzieniszewski, Mikom, Warszawa 2001.
[4] S. Abiteboul, *Querying semi-structured data*, W: Proceedings of ICDT, 1997.
[5] P. Buneman, *Semistructured data*, W: Proceedings of PODS, 1997.
[6] D. Suciu, *Semistructured Data and XML*, W: Proceedings of International Conference on Foundations of Data Organization, 1998.
[7] P. Buneman, S.Davidson, M. Fernandez, D. Suciu, *Adding Structure to Unstructured Data*, W: Proc. of the ICDT, 1997.
[8] K. Wang, H. Liu, *Discovering Typical Structures of Documents: A Road Map Approach*, W: 21st Annual ACM SIGIR Conference on Information Retrieval, s. 146-154, 1998.
[9] W. Waloszek, *Wywodzenie schematu bazy danych na podstawie danych o nieznanej strukturze*, Gdańsk, 2003.
[10] http://xml.gsfc.nasa.gov/archive/
[11] S. McGrath, *XML by Example: Building E-Commerce Applications*, Prentice Hall PTR, 1998.
[12] http://www.w3c.org/XML/Schema
[13] http://www.w3c.org/XML

DERIVING SCHEMAS FROM DATA OF UNKNOWN STRUCTURE USING COBWEB S+T ALGORITHM

Summary

The paper presents COBWEB S+T algorithm for deriving schematic information from sets of XML documents. COBWEB algorithm has been adopted to type and schema clustering, with special functions evaluating precision of sets of XML types and XML schemas. Results of tests of COBWEB S+T algorithm with proposals of future work are presented.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Krzysztof Nowicki*, Tadeus Uhl**

***Politechnika Gdanska, Katedra Systemów Informacyjnych, Polen**
****Flensburg University of Applied Sciences, Institute of Communications Technology, Germany**

PROTOKOLLANALYSE BEI VoIP-SYSTEMEN

Kurzfassung

Der Schwerpunkt dieser Arbeit liegt in den praxisbezogenen Aspekten bei Multimediakommunikation über IP-Transportplattformen. Im ersten Teil der Arbeit werden die in einer realen IP-Umgebung gemessenen Protokollabläufe graphisch dokumentiert und ausgewertet. Der zweite Teil der Arbeit wird den Dimensionierungsaspekten bei der Planung des VoIP-Dienstes in IP-Netzwerken gewidmet. Dabei werden die wichtigsten Parameter für diesen Dienst, d.h. jitter delay, communication patters, packet loss, in Betracht gezogen. Aus dieser Analyse resultierende Erkenntnisse können als Hinweise für die Planer der IP-Netzwerke dienen.

1. EINFÜHRUNG

Die technischen Grundlagen für Voice over IP (abgekürzt VoIP) sind geschaffen worden. Die seit einigen Jahren veröffentlichten Protokolle wie H.323 [1], SIP [2] und MEGACO [3] ermöglichen heutzutage eine paketbasierende Multimediakommunikation. Auf dem Markt sind mehrere VoIP-Systeme (z.B: HiPath 5500 und OpenScape von Siemens, Innovaphone IP von TLK Computer GmbH, Gateways von Cisco, Lucent, Apolio, Selsius usw.) zu haben, die gemäß den o.g. Standards arbeiten. Diese Kommunikationsart findet heute bemerkenswerte Akzeptanz. Dadurch sind neue innovative Dienste, wie z.B. Teleteaching, Telelearning möglich [4].

Das Internet und die meisten Intranetsysteme sind eigentlich nicht für eine Übertragung von zeitkritischen Datenströmen geeignet. Laufzeitdifferenzen zwischen Paketen und die Paketverluste gehören zu den markantesten Eigenschaften eines IP-Netzwerkes und stellen für die Audio- und Videokommunikation große Probleme dar. Es ist also sinnvoll, den Einfluss von den o.g. Parametern auf die Quality of Service (QoS) bei VoIP zu untersuchen. Daraus resultierenden Erkenntnisse sollten dann bei der Dimensionierung von IP-Netzen für VoIP Anwendungen verwendet werden. Dieser Problematik wird diese Arbeit gewidmet.

2. PROTOKOLLANALYSE IN VoIP-UMGEBUNG

Die Referenzkonfiguration der VoIP-Architektur kann aus den ITU-T Empfehlungen (siehe [5]) entnommen werden.

2.1. Testumgebung

Die verwendete bei Untersuchungen VoIP-Testumgebung zeigt die Abb. 1 [6]. Der Protokollanalysator ist direkt an das „shared" LAN angeschlossen und ist somit in der Lage sämtlichen Datenverkehr auf dem Netz aufzuzeichnen. Über den gleichen Port kann auch eine definierte Last von dem integrierten Lastgenerator auf das Netz gesendet werden. Diese Last dient zur Beeinflussung der zu messenden Daten. Mit diesem Gerät ist es somit möglich, neben einer Verkehrsmessung gleichzeitig Last zu erzeugen und die Netzaktivitäten in Echtzeit anzuzeigen.

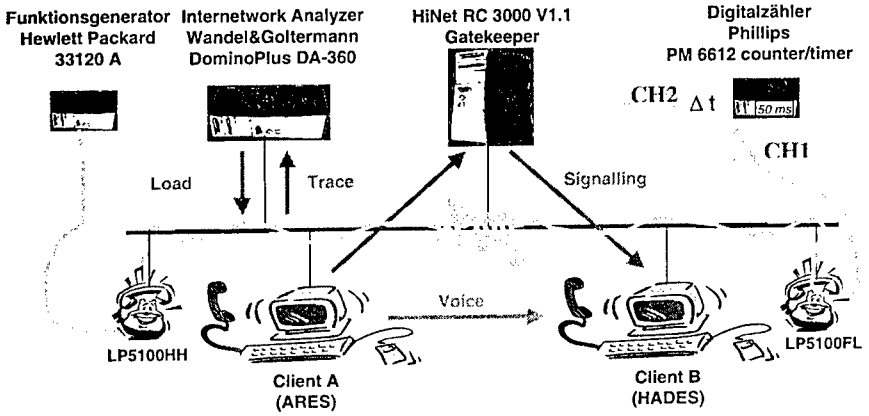


Abb. 1.: Der verwendete VoIP-Testaufbau

2.2. Funktionelle Analyse

In den Abb. 2 und 3 wird die Signalisierungsphase bei dem Aufbau einer Verbindung mit einem Gatekeeper dargestellt [6]. Durch das RAS (H.225.0)-Protokoll wird die Registrierung und der Zugang der einzelnen Terminals geregelt, sowie die Bandbreitenkontrolle übernommen. Der eigentliche Verbindungsaufbau erfolgt gemäß dem H.225.0 Protokoll. Dies wird mit dem Befehl Setup eingeleitet und mit der Nachricht Alerting abgeschlossen.

Sobald Client B das Gespräch annimmt, sendet er ein Connect über den Gatekeeper zum Client A mit der Information über welchen Port der Aufbau des logischen Kanals erfolgen soll (siehe Abb. 3). In diesem Fall werden die Ports für Client A und B einfach nur „hochgezählt" und der logische Informationskanal wiederum über das „TCP-Three-Way-Handshake-Verfahren" gesichert. Hierüber läuft dann die Verbindungssteuerung mittels des H.245-Protokolls ab, wo Steuerungsfunktionen, wie Aushandlung von Endgerätefunktionen (Video, Audio, Daten), Steuerung der logischen Kanäle, Master-Slave-Erkennung und Flusskontrolle stattfindet.

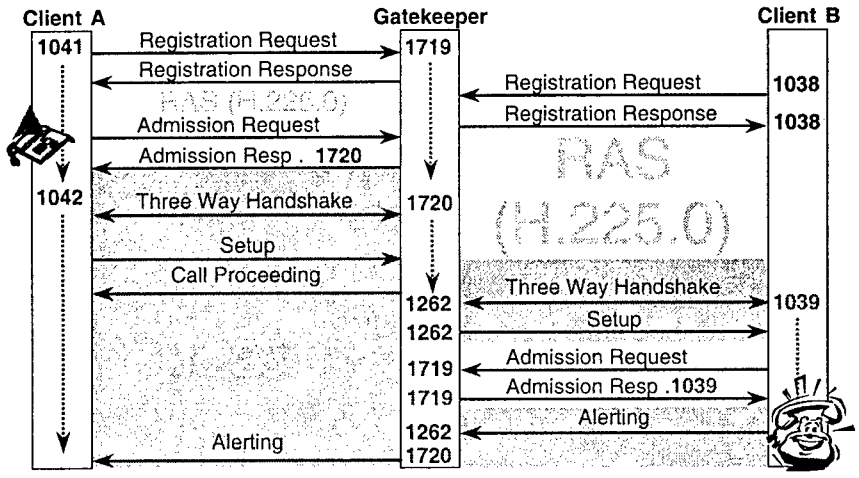


Abb. 2: Die Signalisierungsphase einer H.323-Verbindung mit Gatekeeper (Teil1)

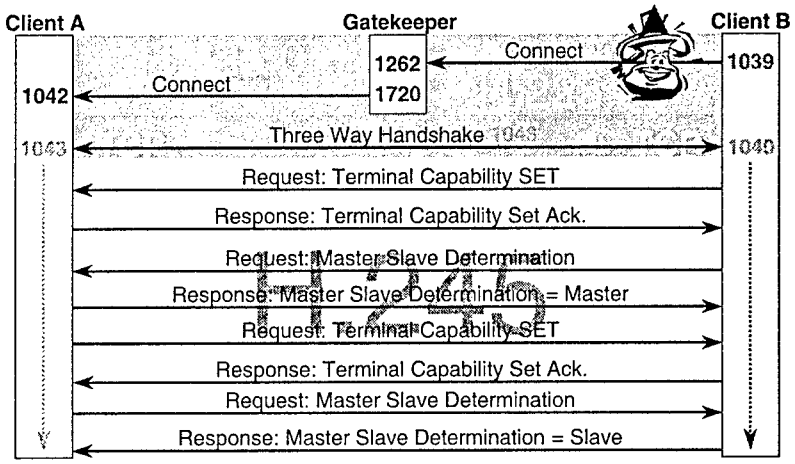


Abb. 3: Signalisierungsphase einer H.323-Verbindung mit Gatekeeper (Teil2)

Nach der Klärung dieser Verwaltungsfragen zwischen Client A und Client B werden die logischen Kanäle für den Austausch der Sprachpakete zwischen den beiden Clients geöffnet, indem sie sich die RTP- und RTCP-Portnummern für die Adressierung gegenseitig austauschen (vgl. Abb. 4).

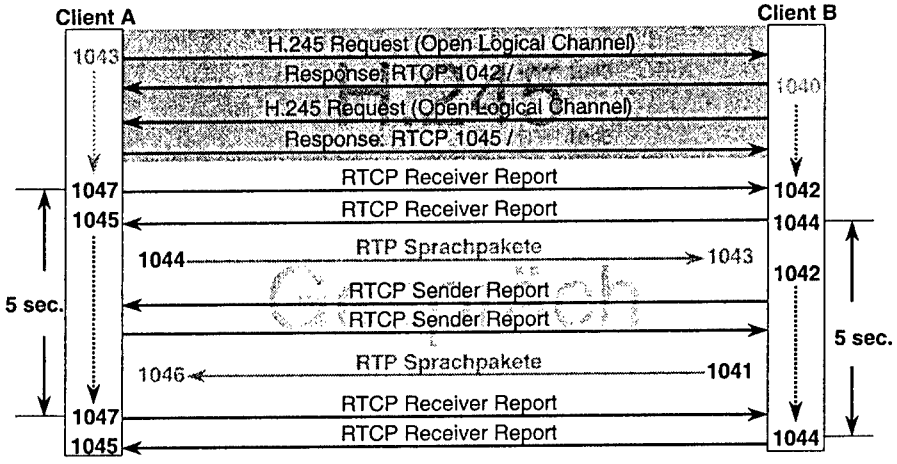


Abb. 4: Gesprächphase bei einer H.323-Verbindung

Während der Gesprächphase werden die Sprachdaten als RTP-Pakete in beiden Richtungen regelmäßig ausgetauscht. Die RTCP-Pakete (gesendet in den Abständen von 5 Sekunden) dienen der Qualitätskontrolle der Sprachverbindung.

Die Verbindungsabbauphase wird mit der Nachricht Request Close Logical Channel eingeleitet und an den Gesprächpartner gesendet (siehe Abb. 5). Der wiederum verschickt ein Response als Bestätigung und der H.245-Verbindungssteuerungskanal wird durch ein End Session Command mit dem Inhalt Disconnect abgebaut. Der gleiche Client, der für den Abbau des H.245-Kanals gesorgt hat, baut ebenfalls die RAS-Verbindung zu seinem Verbindungspartner über den Gatekeeper ab und sendet ein H.225.0 Release Complete.

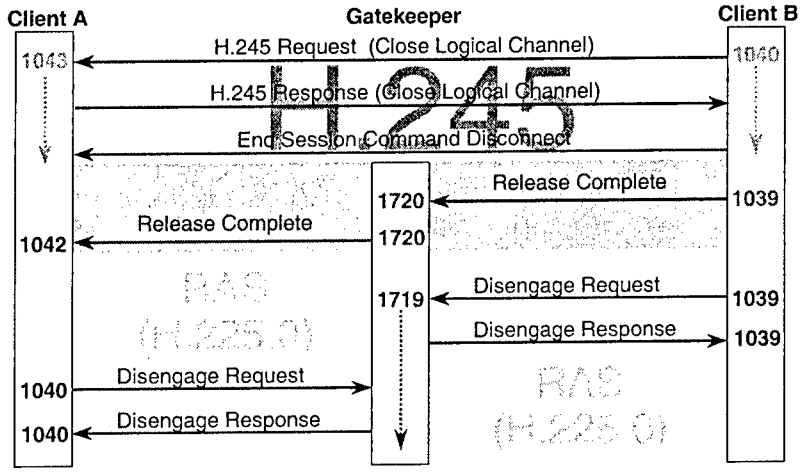


Abb. 5: Verbindungsabbauphase einer H.323-Verbindung

Für den Abbau der RAS-Verbindung sorgen die Clients, indem sie ein Disengage Request an den Gatekeeper senden und somit den Beendigungswunsch verdeutlichen. Die Portnummern, die bei dem Verbindungsabbau verwendet werden, entsprechen den Portnummern des Verbindungsaufbaus. Und zwar liegt es daran, dass keine der genannten Verbindungen vorher, z.B. während der Gesprächsphase, abgebaut wurde. Jedes Protokoll wird in umgekehrter Reihenfolge des Verbindungsaufbaus bei Beendigung des Gespräches abgebaut, so dass es für eine erneute Verbindung zur Verfügung steht.

2.3. Paketanalyse

Als nächstes werden die RTP-Sprachpakete einer Analyse unterzogen. Hierbei geht es um die Untersuchung des Aufbaus dieser Pakete. Dabei ist neben dem reinen Nutzdatenteil ein Datenoverhead (verschiedene Header) zu unterscheiden. Dieser Datenoverhead setzt sich folgendermaßen zusammen: Ethernet-Header = 14 Byte + IP-Header = 20 Byte + UDP-Header = 8 Byte + RTP-Header = 16 Byte. Insgesamt ergibt sich ein Overhead von 58 Byte. Zieht man den Datenoverhead von den reell gemessenen Paketgrößen ab, so erhält man die sog. Payload Size, wie in Abb. 6 dargestellt. Die Payload Size wiederum entspricht dem ganzzahligen Vielfachen der Frame Size und setzt sich somit aus einer beliebigen Anzahl von Frames zusammen. In diesen Frames befinden sich die reinen Sprachdaten.

Im weiteren wird folgende Größe definiert:

$$Frame\ Duration\ \ [s] = \frac{Frame\ Size\ \ [bit]}{\ddot{U}bertragungsrate\ der\ Sprache\ \ [bps]}$$

Mit der Frame Duration ist also die Zeit gemeint, die der Wiedergabedauer des Dateninhaltes eines Paketes entspricht. Sie ist vom Art des verwendeten Codec abhängig (vgl. Abb. 6).

Senderichtung	Codec	Packet Size [Byte]	Payload Size [Byte]	Frames / Packet	Frame Size [Byte]	Frame Duration [ms]
Ares Hades	G.711	314	256	1	256	32
LPFL Ares	G.711	298	240	1	240	30
Ares LPFL	G.711	314	256	1	256	32
Ares LPHH	G.711	314	256	1	256	32
LPFL LPHH	G.711	298	240	1	240	30
LPHH Ares	G.711	538	480	2	240	30
LPHH LPFL	G.711	538	480	2	240	30
LPHH LPFL	G.723.1	68	10	1	10	12,5
LPFL LPHH	G.723.1	82	24	1	24	30

Abb. 6: Berechnung der Frame Size und Frame Duration

Die Abb. 7 zeigt den signifikanten Unterschied des Verhältnisses von Nutzdaten zu Gesamtdaten bei der Senderichtung *LPHH→LPFL* der beiden IP-Telefone unter Verwendung verschiedener Audio Codecs.

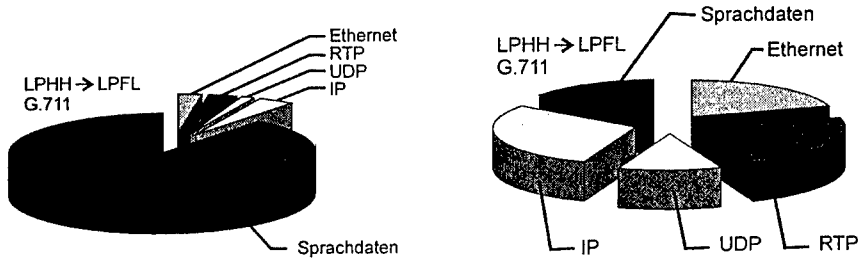


Abb. 7: Sprachdaten im Ethernetpaket

Um die Analyse fortzusetzen, wird der Begriff Effizienz der Übertragung eingeführt. Sie wird wie folgt definiert:

$$Effizienz\ der\ Übertragung \quad [\%] = 100 \frac{Payload\ Size \quad [Byte]}{Packet\ Size \quad [Byte]}$$

Die Abb. 7 zeigt, dass sich im Fall des Codec G.711 eine Effizienz der Übertragung von 90% und im Fall des Codec G.723.1 eine Effizienz von nur 15% ergibt. Eine Verbesserung der Effizienz würde man erzielen, wenn man die Payload Size, also den Nutzdateninhalt, erhöhen würde. Die Verbesserung würde man dann aber mit einem höheren Delay bezahlen, da das Paketierungsdelay proportional zur Payload Size ansteigt.

Die Angaben aus der Abb. 7 lassen zu, eine weitere wichtige für die Praxis Größe zu bestimmen. Diese Größe heißt benötigte Bandbreite pro Verbindung und wird wie folgt definiert:

$$Benötigte\ Bandbreite \quad [bps] = \frac{Packet\ Size \quad [bit]}{Frame\ Duration \quad [\text{sec}]}$$

Die Abb. 8 zeigt charakteristische Parameter der untersuchten VoIP-Verbindungen.

Senderichtung	Codec	kbps	Packet Size [Byte]	Packets/ Sec	BB kbps	% Effizienz	% 2* BB/ 10 Mbps	Delay [ms]
Ares Hades	G.711	64	314	31,25	7,85	82%	1,57	32,00
LPFL Ares	G.711	64	298	33,33	7,94	81%	1,59	30,00
Ares LPFL	G.711	64	314	31,25	7,85	82%	1,57	32,00
Ares LPHH	G.711	64	314	31,25	7,85	82%	1,57	32,00
LPFL LPHH	G.711	64	298	33,33	7,94	81%	1,59	30,00
LPHH Ares	G.711	64	538	16,67	7,17	90%	1,43	60,00
LPHH LPFL	G.711	64	538	16,67	7,17	90%	1,43	60,00
LPHH LPFL	G.723.1	6,4	68	80,00	4,35	15%	0,87	12,50
LPFL LPHH	G.723.1	6,4	82	33,33	2,18	30%	0,44	30,00

Abb. 8: Berechnung der Bandbreite und des Paketierungsdelys

Wie in der Abb. 8 zu sehen ist, entspricht das Delay der Verbindung LPHH → LPFL bei G.711 dem doppelten normalen Delay von 30 ms. Dieses liegt daran, dass zwei Frames in einem RTP-Paket eingebettet sind. Deshalb wird durch die Übertragung auch nur eine Bandbreite von 1,43% bezogen auf 10 MBit/s, im Gegensatz zu den sonst üblichen 1,59%, in Anspruch genommen.

3. DIMENSIONIERUNGSASPEKTE BEI VOIP

3.1. Jitter-Theorie

Der Begriff Jitter steht für Schwankungen der Zwischenankunftszeiten von Paketen. Der Jitter darf in der Regel einen maximalen Wert nicht überschreiten, um eine Übertragung von isochronen Diensten zu ermöglichen. Wichtig ist nicht nur der Absolutwert des Jitters sondern auch der zeitliche Verlauf dieser Leistungsgröße und die Häufigkeit des Auftretens der einzelnen Jitter-Werte.

Zur Darstellung des Jitters sind mehrere Verfahren konzipiert worden [7]. Zu den wichtigsten Darstellungen gehören: a) Einfachband, b) Doppelband und c) Kommunikationsmuster.

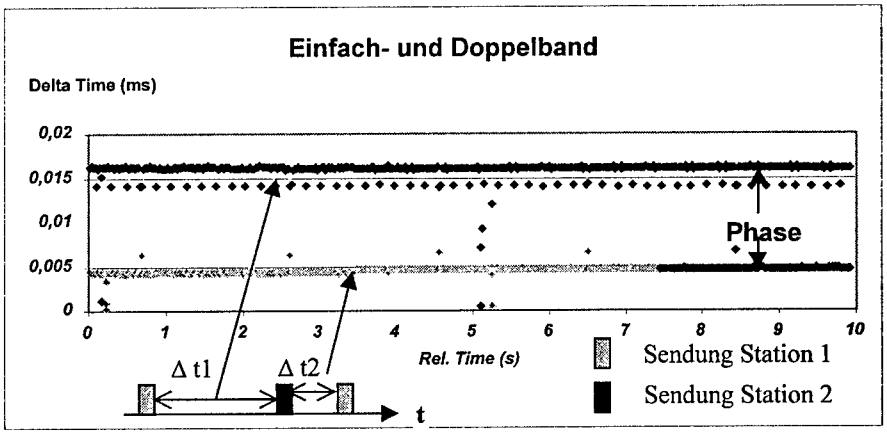


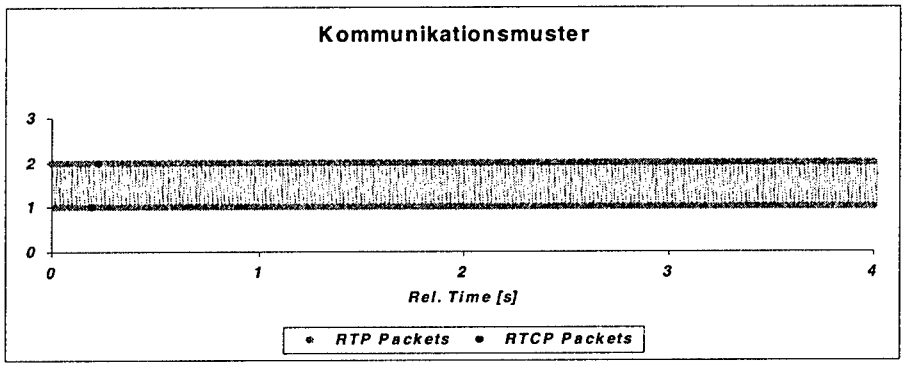
Abb. 9: Illustration zum Einfach- und Doppellband

Das Einfachband stellt die Ankunftszeiten zwischen Paketen (Inter arrival Time) bezogen auf die von einer Station ausgesendeten RTP-Pakete dar, und wird darum als Station Jitter bezeichnet (vgl. Abb. 9).

Das Doppelband bezieht sich auf die gemeinsame Darstellung der Inter arrival Times zweier Stationen, und wird darum als Connection Jitter bezeichnet (vgl. Abb. 9).

Der Abstand zwischen den beiden Bändern wird als Phase bezeichnet. Addiert man die Zeiten, auf die sich die beiden Bänder beziehen, erhält man die Station Jitters beider Stationen. Sind keine Linien zu erkennen, ist dies ein Indiz für eine qualitätsmäßig schlechte Kommunikationsverbindung.

Eine weitere Möglichkeit zur Darstellung einer Verbindungsübersicht zweier Stationen stellt das Kommunikationsmuster dar (vgl. Abb. 10).



Das Prinzip zur Erstellung des Kommunikationsmusters lautet: Die Ziffern auf der Y-Achse symbolisieren die beiden beteiligten Stationen, auf der X-Achse läuft der Zeitstrahl. Wird die Kommunikationsrichtung gewechselt, entsteht eine Verbindungslinie zwischen den beiden horizontal laufenden Stationslinien und es ist ein optisches Muster zu erkennen.

Wenn die Stationen nacheinander ihre Sprachpakete senden, ergibt sich ein gut ausgefülltes Muster, was auf gute Qualität der Verbindung deutet. Sind Störungen oder sog. „Halbwege" vorhanden, lässt sich das anhand von Lücken (Drop Outs) im Kommunikationsmuster nachweisen. Halbweg beschreibt den Zustand, wenn nur noch eine Station sendet. Dies tritt häufig in extremen Lastsituationen auf.

3.2. Bandbreite bei H.323-Verbindung

Der im Punkt 2.2 dargestellte Verbindungsaufbau einer H.323-Verbindung mit Gatekeeper wurde auf die genutzte Bandbreite hin untersucht. Das Ergebnis dieser Untersuchung wird in Abb. 11 dargestellt.

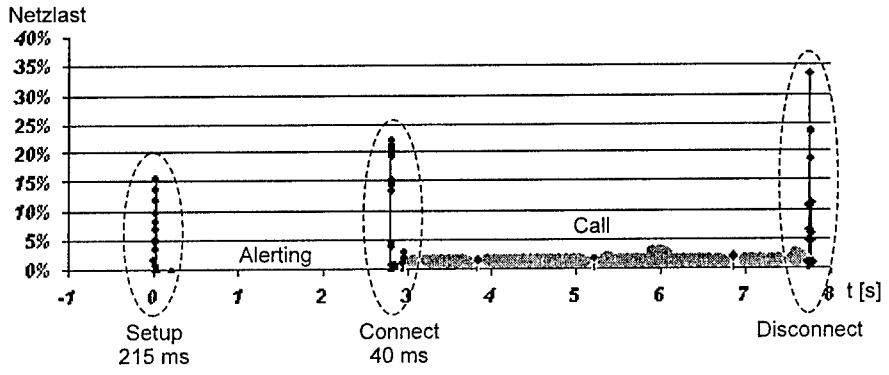


Abb. 11: Bandbreite bei einer H.323-Verbindung

Hier werden die beiden Bereiche der Signalisierung und der Gesprächsphase graphisch unterschieden. Die schwarzen Signalisierungsabläufe weisen eine Peak-Last bis zu 35% auf, wobei die hell grau dargestellte Gesprächsphase lediglich einen Bandbreitenbedarf von ca. 1,6 % aufweist. Diese Prozentzahlen beziehen sich auf eine Übertragungsrate von 10 Mbps. Dadurch konnten die theoretischen Berechnungen im Punkt 2.3 bestätigt werden.

Bei der Beobachtung, dass die Signalisierung in allen Phasen eine relativ hohe Peak-Last erzeugt, ist zu bedenken, dass sie durch das gesicherte Transport Control Protocol übertragen wird und somit nicht in Echtzeit erfolgen muss. Dies bedeutet, dass nur die Bandbreite genutzt wird, die zur Verfügung steht, im Umkehrschluss also auch bei minimaler Bandbreite ein gesicherter Verbindungsaufbau stattfinden kann.

3.3. Qualitative und quantitative Analyse

Zuerst wird auf die qualitative Bewertung der Sprachqualität bei VoIP durch subjektive Höreindrücke von Testpersonen eingegangen. Hierbei werden für die Beurteilung der Sprachqualität drei Bereichen unterschieden:

A: bestmögliche Qualität,
B: gestörte Kommunikation,
C: Verständigung unmöglich.

Innerhalb des Bereiches A war es den Testpersonen nicht möglich, einen weiteren Qualitätsunterschied zu klassifizieren. Dies liegt an der Tatsache, dass kleinere Jitter-Werte durch den vorgehaltenen Jitter-Buffer ausgeglichen werden. Wird dieser Bereich überschritten, so kommt es zu einer sprunghaften Verschlechterung der Sprachqualität (z.B. im Bereich B durch Halbwege) und bei weiterer Überschreitung letztlich zu einer vollständig gestörten Kommunikation (Bereich C).

In der Abb. 12 werden die Bereiche der quantitativen und der qualitativen Messungen zusammengeführt. Wie aus der Abb. 12 ersichtlich, wurde bei den Untersuchungen sowohl Paketgröße für die Netzwerklast als auch die Netzlast selbst variiert. Die Ergebnisse zeigen, dass eine Beeinflussung der Sprachqualität erst ab einer Belastung des Netzes von 90% festzustellen ist. Für größere Pakete im Hintergrund des Netzes vergrößert sich diese Grenze.

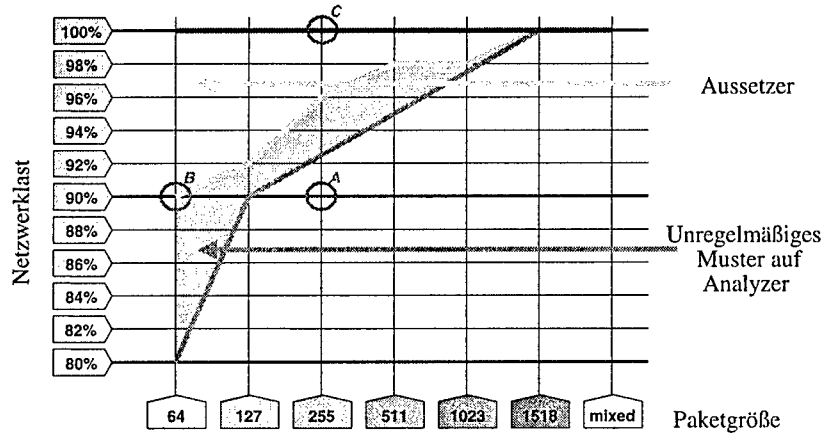


Abb. 12: Matrix mit qualitativer und quantitativer Analyse

Die Abb. 12 zeigt, dass die Ergebnisse der Sprachqualitätsanalyse durch die quantitative Analyse voll gestützt werden. Es hat sich ein Bereich gezeigt, in dem die quantitative Analyse bereits greift, wo die Wahrnehmungsfähigkeit des menschlichen Ohres noch nicht ausreicht. Dieser schmale Bereich könnte als „Puffer" dafür genutzt werden, um mit den geeigneten Mitteln auf Zustände des Netzwerkes und letztendlich des Gespräches geeignet zu reagieren, also Gegenmaßnahmen zu treffen, bevor eine Verschlechterung der Sprachqualität eintritt.

3.4. Dimensionierungsregel

Resümierend die Erkenntnisse aus den Untersuchungen lässt sich folgendes sagen:

- Bei einer Hindergrundlast von 0% – 90% ist das VoIP-System stabil.
- Bei einer Hindergrundlast größer als 90% ist eine Kommunikation kaum möglich.
- Größere Pakete im Hintergrund begünstigen eine VoIP-Kommunikation.
- Bei Sprachkommunikation weist der Verkehr den CBR (Constant Bit Rate) Charakter auf (Ausnahme Netmeeting-Applikationen).
- Eine Sprachverbindung mit dem Codec G.711 nimmt ca. 160 kbps der Bandbreite in Anspruch.
- Eine Sprachverbindung mit dem Codec G.723.1 nimmt ca. 80 kbps der Bandbreite in Anspruch.
- Peak-Last bei Signalisierung kann kurzfristig bis zu 35 % der Netzbandbreite in Anspruch nehmen.
- Soll Effektivität der Übertragung erhöht werden (mehrere Frames in einem Paket), muss Kapazität des Ausgleichs-Puffers entsprechend dimensioniert werden.

Unter Berücksichtigung der o.g. Feststellungen lässt sich bei der Multimedia-kommunikation in einem IP-Netz folgende Dimensionierungsregel angeben:

Benötigte Gesamtbandbreite muss $\leq$ als die im Netz zur Verfügung stehende Bandbreite sein.

$$\textit{Benötigte Gesamtbandbreite} = \frac{\textit{Min BB für Sprache} + \textit{Min BB für Video} + \textit{Min BB für Daten}}{0{,}75}$$

Der Faktor 0,75 berücksichtigt den zusätzlichen Verwaltungsverkehr in der IP-Transportplattform. Bei der Bestimmung der minimalen Bandbreite pro Dienst muss die Anzahl der für den Dienst vorgesehenen Clients berücksichtigt werden. Oder umgekehrt, ausgehend aus der im Netz zur Verfügung stehenden Bandbreite (Sie kann durch Messungen ermittelt werden.) lässt sich maximale Anzahl der Clients pro Dienst ermittelt. Dabei ist es zu Beachten, dass bei jeder Verbindung mindestens zwei Clients beteiligt sind. Diese 75%-Regel wurde in vielen Experimenten angewandt und hat sich als sehr geeignet für die Praxis bestätigt.

4. ZUSAMMENFASSUNG

Im Rahmen dieser Arbeit wurde das Thema Protokollanalyse bei VoIP aus der theoretischen und praktischen Sicht näher behandelt. Die erhaltenen Ergebnisse lieferten Hinweise für die Dimensionierung von IP-Netzwerken mit den Multimediadiensten.

Aus den Untersuchungserkenntnissen wurde eine Dimensionierungsregel erarbeitet, die heißt: Die benötigte Gesamtbandbreite für die Multimediadienste dividiert durch den Faktor 0,75 soll im Anklang mit der in einem IP-Netzwerk zur Verfügung stehenden Bandbreite gebracht werden. Diese einfache Regel wurde durch mehrere Experimente in den IP-Netzwerken bestätigt und scheint ein geeignetes Werkzeug bei der Planung von VoIP-Diensten zu sein.

Die im Rahmen dieser Arbeit verwendeten Netzlasten waren relativ einfach und könnten im weiteren noch komplexer behandelt werden. Vor allem wäre interessant, mit Multimedia-Lastgeneratoren im Netzhintergrund zu arbeiten. Ebenfalls wünschenswert wäre es, Untersuchungen der QoS Parameter im Falle mehrerer gleichzeitig aktiver VoIP-Verbindungen durchführen zu können. Das alles hätte die in dieser Arbeit vorgestellten Ergebnisse gut ergänzt. In diese Richtung sollten weitere Arbeiten geplant werden.

5. REFERENCES

[1] http://www.openh323.org/standards.html
[2] http://www.sipcenter.com/files/sip2.pdf
[3] http://www.ietf.org/ftml.charters/megaco-charter.html
[4] Rektor der Technischen Universität Ilmenau (Herausgeber): Multimedia für Bildung und Wirtschaft. Tagungsband zu dem 7. Workshop MBW, Ilmenau, 2003
[5] Nowicki K., Woźniak J.: Przewodowe i bezprzewodowe sieci LAN, OWPW, Warszawa 2002.
[6] Clausen, C., Schmidt, S.: Multimediale Kommunikationslösungen über IP-Netzwerke. Diplomarbeit, FH Flensburg, Institut für Kommunikationstechnologie, 2000
[7] Uhl, T.; Zäncker, O.: Voice over IP exemplifed by Siemens HiNet RC 3000. Proc. of the ICC-2001, St. Petersburg/Russland, Juni 2001.

PROTOCOL ANALYSIS OF VoIP-SYSTEMS

Abstract

The focus of this paper lies in the practical aspects of multimedia communications over IP transport platforms. A detailed description of protocol procedures in multimedia communication, measured in a real IP environment, will then be presented. The protocol analyses that were conducted in course of the investigation will be documented in diagrams and evaluated. A further part of the paper devotes itself to the question of dimensioning at the planning stage of the VoIP service in IP networks and deals with the service's most important parameters, i.e. jitter delay, communication patters and packet loss. The insights gained from the investigation can be considered as useful guidelines for the architecture of IP networks.

ANALIZA SYSTEMÓW VOIP

Streszczenie

W artykule rozważono praktyczne aspekty komunikacji multimedialnej w sieciach IP. Opisano zrealizowany system testowy VoIP. Dokonano szczegółowej prezentacji (diagramy czasowe) procedur protokolarnych realizowanych w zbudowanym do celów badawczych środowisku testowym VoIP. Oceniono wpływ doboru kodeka, długości pakietów, obciążenia sieci na jakość realizacji usług VoIP. Przedstawiono propozycje reguł projektowania systemów VoIP-H.323.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Lech Hasse[1], Marek Illeczko[2]

[1]Katedra Metrologii i Systemów Elektronicznych, Politechnika Gdańska
[2]Przedsiębiorstwo Innowacyjno-Wdrożeniowe MERITUM, Toruń

AKWIZYCJA I PRZETWARZANIE DANYCH W SYSTEMIE TELEMONITORINGU SIECI CIEPŁOWNICZEJ

Streszczenie

Przedstawiono wymagania funkcjonalne systemu automatyki i zdalnego nadzoru rozległej sieci ciepłowniczej. Opisano funkcje oprogramowania systemowego dla modułów komunikacyjnych i dla modułów funkcjonalnych oraz metodykę tworzenia i zarządzania baza danych systemu. W komunikacji dla zarządzania systemem automatyki zastosowano standard komunikacji LonWorks z protokołem LonTalk. Opisano reprezentatywne węzły sieci telemetrycznej i stosowane rodzaje komunikacji. Przedstawiony w pracy system został wdrożony dla potrzeb PEC Toruń

1. ZADANIA SYSTEMU AUTOMATYKI I NADZORU SIECI CIEPŁOWNICZEJ

Kompleksowa modernizacja sieci ciepłowniczych oraz wprowadzenia systemów automatycznego sterowania ich parametrami i zabezpieczeniami spowodowały konieczność wprowadzenia systemów ich telemonitoringu, a więc zastosowanie w systemach elementów, urządzeń oraz oprogramowania współczesnej automatyki i informatyki. Istotnym elementem zapewnienia poprawnej pracy całego systemu jest właściwa analiza zebranych informacji prowadząca do racjonalnego zarządzania siecią.

Problemy, które należy przeanalizować i rozwiązać dla danej sieci ciepłowniczej, to zapewnienie:

- skuteczności hydraulicznej i cieplnej,
- bezpiecznych i optymalnych warunków pracy źródeł ciepła,
- prawidłowej i szybkiej reakcji systemu na stany awaryjne,
- możliwie najniższego poziomu kosztów przesyłu i rozdziału ciepła.

Rosnący udział w pełni zautomatyzowanych węzłów cieplnych, znaczne różnice między wartością rzeczywistego a prognozowanego zapotrzebowania na moc cieplną obiektów, a także występowanie nadmiarów mocy cieplnej w źródłach ciepła, stwarzają duże wymagania na warunki pracy systemu. Istotnym założeniem przy jego tworzeniu jest przyjęcie systemu otwartego, bazującego na najnowszych rozwiązaniach technicznych i charakteryzującego się otwartą architekturą, standardową platformą sprzętową, pro-

tokołami i procedurami dającymi użytkownikowi możliwość wyboru produktów i elementów systemu z dużej liczby dostępnych na rynku rozwiązań przy jednoczesnej możliwości ich integracji.

W opracowanym systemie telemonitoringu sieci ciepłowniczej można wyróżnić następujące funkcje i uprawnienia użytkowników systemu:

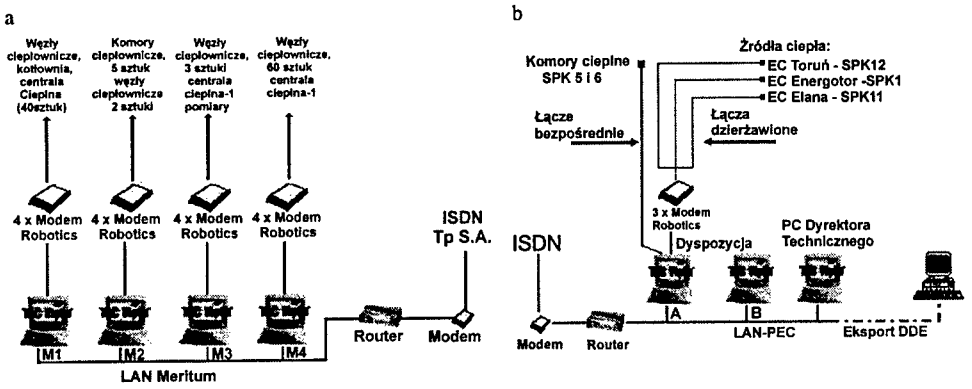
- funkcje podstawowe – w skład oprogramowania podstawowego wchodzą:
  - system kontroli dostępu – umożliwia korzystanie z systemu tylko osobom upoważnionym; możliwości operatora określają *poziom dostępu, kategoria i uprawnienia,*
  - system obsługi alarmów,
  - moduł grafiki kolorowej,
  - funkcje czasowe używane do sterowania urządzeń oraz do aktywowania różnych funkcji w systemie,
  - dokumentacja systemu,
- rejestracja zdarzeń – automatyczne gromadzenie danych z monitorowanych obiektów, wykorzystywanych następnie do tworzenia raportów i prezentacji. Moduł rejestracji zdarzeń, zainstalowany na poszczególnych stanowiskach operatorów, aktywuje rejestrację w sterownikach oraz cyklicznie odczytuje dane ze sterowników. Dalsze przetwarzanie tych danych dokonywane jest przez moduły generatora raportów i prezentacji,
- rejestracja historii – zapisywanie i przechowywania danych o zdarzeniach w systemie,
- generator raportów – oddzielny program, umożliwiający operatorowi tworzenie dokumentacji stanu nadzorowanej instalacji, wartości mierzonych, statusów alarmów,
- generator prezentacji danych – program używany w celu edycji i prezentacji danych zgromadzonych za pomocą modułu rejestracji,
- edytor kolorowej grafiki – umożliwia tworzenie i edycję rysunków zawierających informacje zarówno statyczne, jak i dynamiczne,
- generator bazy danych – narzędzie do zbierania, zarządzania oraz programowania informacji generowanych w systemie.

2. OPIS STRUKTURY SYSTEMU

Koncepcję budowy systemu oparto na produktach firmy TAC [1]. Podstawowymi składnikami systemu są: oprogramowanie TA Vista, sterowniki swobodnie programowane TA Xenta, środki łączności (w tym rozwiązaniu dzierżawione, komutowane oraz ISDN).

TA Vista jest narzędziem zawierającym moduły komunikacyjne i funkcyjne. Dostosowanie programu TA Vista do konkretnych wymagań, polega na stworzeniu bazy danych, zawierającej obiekty odwzorowujące nadzorowane elementy systemu (czujniki pomiarowe, siłowniki, pompy itp.) oraz obiekty związane z organizacją pracy systemu (grafika, obsługa alarmów, uprawnienia użytkowników itp.). Modułowa budowa programu oraz sterowników umożliwia optymalne dostosowanie ich do potrzeb oraz zapewnia prostą rozbudowę. Oprogramowanie zapewnia pełną komunikację, łącznie z automatycznym wydzwanianiem w przypadku wystąpienia alarmu i nie wymaga dodatkowych modułów komunikacyjnych. Do lokalnej obsługi sterowników przewidziano panele operatorów. Schemat blokowy struktury sieci LAN dla sieci ciepłowniczej Torunia, zarówno od strony

stacji dyspozytorskiej użytkownika, jak i od strony stanowiska serwerów w MERITUM, z uwzględnieniem elementów wyposażenia i komunikacji, przedstawiono na rys. 1 [2].



Rys. 1. Uproszczony schemat blokowy komunikacji i wyposażenia systemu:
a – stanowiska serwerów sieci LAN w MERITUM, b – struktura stacji dyspozytorskiej PEC

Ukierunkowanie obiektowe systemu pozwala traktować wszystkie jego elementy jako obiekty. Programy aplikacyjne TAC Vista zawierają bazę danych całego systemu. Jest ona rozdzielona na poszczególne stanowiska, z uwzględnieniem funkcji jakie pełnią w systemie.

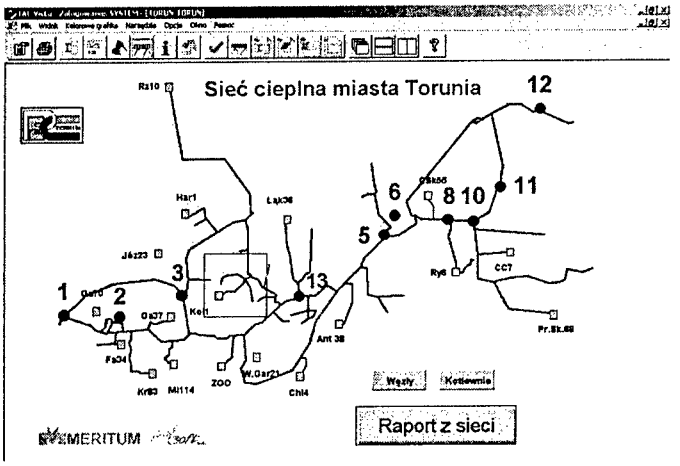
3. BAZA DANYCH SYSTEMU

Komputer w centralnej dyspozytorni pełni rolę komputera głównego sieci. Jego baza danych zawiera wszystkie obiekty globalne systemu:

- uprawnienia użytkownika – obiekt w którym są zdefiniowane identyfikatory, hasła
  i kompetencje wszystkich użytkowników systemu,
- suma alarmów optycznych i akustycznych – obiekty aktywowane przy dowolnym alarmie w systemie z określonym priorytetem (różnią się one sposobem potwierdzania).

Ponadto w komputerze głównym zdefiniowane są obiekty zawierające informacje o zasobach fizycznych pozostałych stanowisk operatorskich w sieci. Baza danych komputera głównego zawiera wszystkie obiekty graficzne systemu oraz formularze raportów używanych przez administratora systemu.

Komputer w centralnej dyspozytorni, przeznaczony do obsługi alarmów, posiada bazę danych zawierającą obiekty odpowiedzialne za obsługę alarmów w systemie. Baza danych komputera w dyspozytorni centralnej na stanowisku dyspozytora zawiera obiekty reprezentujące wszystkie zasoby urządzeń fizycznie podłączonych do tego stanowiska, obiekty graficzne całego systemu oraz formularze raportów używanych przez dyspozytora. Dodatkowo stworzony jest obiekt graficzny, zawierający mapę sieci ciepłowniczej (rys. 2) z naniesionymi źródłami ciepła i głównymi komorami ciepłowniczymi, powiązany dynamicznie parametrami odzwierciedlającymi aktualny status pracy urządzeń.



Rys. 2. Ogólny schemat sieci ciepłowniczej (na przykładzie miasta Torunia) widziany ze stanowiska dyspozytora

Do tworzenia i zarządzenia bazą danych w TAC Vista służy dodatkowy moduł programowy TAC Vista Explorer. Tworzenie bazy danych polega na wyborze z menu kontekstowego wymaganego obiektu, jego skonfigurowanie, a następnie „wklejenie" pliku konfiguracyjnego sterownika utworzonego w TAC Menta. Dla użytkownika – operatora systemu, tworzony jest zbiór skrótów zmiennych sterownika, które są wizualizowane i monitorowane przez TAC Vista.

4. REPREZENTATYWNE WĘZŁY SIECI CIEPŁOWNICZEJ

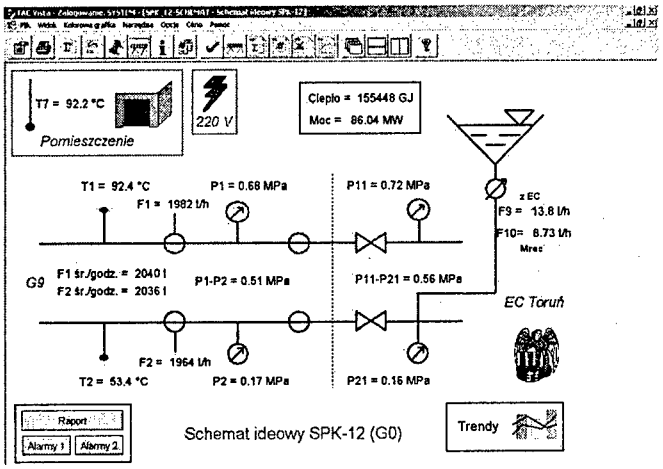
W opracowanym systemie nadzorowane obiekty podzielono na następujące grupy funkcyjne [2]:

- źródła ciepła,
- komory ciepłownicze,
- komory rozgałęźne z telemechaniką,
- bezobsługowe kotłownie olejowe,
- centrale cieplne,
- indywidualne węzły cieplne w budynkach.

Głównym zadaniem źródeł jest kontrola dostarczanych parametrów medium oraz pomiary jego ilości. Na podstawie informacji z tych węzłów, dyspozytor ma możliwość prowadzenia analizy pracy i regulacji całej sieci ciepłowniczej pod względem ilościowo – jakościowym. Pomiary parametrów pozwalają na kontrolę pracy sieci pod względem bezpieczeństwa. W tak newralgicznym punkcie, jakim jest źródło ciepła musi być prowadzona rejestracja wszystkich parametrów, zachowań obsługi oraz alarmów systemowych.

Do wykonania punktów telemetrycznych źródeł ciepła użyto szeregu specjalnie dobranych czujników, a do rejestracji wyników pomiarów oraz ich przesyłu na stanowiska dyspozytora sterowniki swobodnie programowalne TA 6751. Na rys. 3 przedstawiono schemat funkcjonalny przykładowego węzła telemetrycznego [2]. Kontrolowane są następujące parametry węzła:

- temperatury zasilania i powrotu,
- ciśnienia statyczne w rurociągu zasilającym i powrotnym, po stronie sieci,
- ciśnienia statyczne w rurociągu zasilającym i powrotnym po stronie źródła,
- przepływy medium, zarówno w przewodzie zasilającym jak i powrotnym,
- przepływ uzupełnienia wody w źródle charakteryzujący ubytki w sieci.



Rys. 3. Schemat funkcjonalny węzła telemetrycznego SPK-12- EC TORUŃ

Dodatkowo oprócz kontroli parametrów odczytywane są informacje alarmowe pozwalające na ustawienia granic alarmowych mierzonych parametrów, informacje o zaniku zasilania w węźle oraz o temperaturze w kontrolowanym pomieszczeniu.

Do realizacji funkcji węzła użyto sterowników swobodnie programowalnych TAC Xenta 401N/P, modułów rozszerzeniowych TAC Xenta 451, TAC Xenta 421, czujników temperatury, przetworników ciśnienia, siłowników i zaworów regulacyjnych. Dla celów monitoringu wyposażono regulator w aplikację komunikacyjną oraz podłączono go przez modem telefoniczny do systemu.

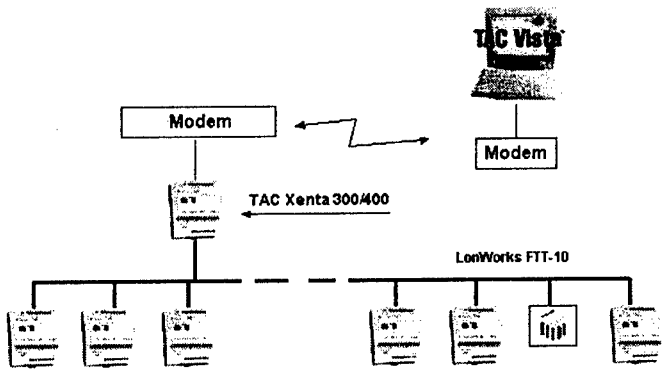
5. KOMUNIKACJA W SYSTEMIE

W opisanym systemie telemonitoringu występują różne rodzaje komunikacji, w tym:
- komunikacja w relacji klient-serwer wykorzystująca protokół DDE,
- komunikacja sieci LonWorks,
- komunikacja modemowa do komunikowania się sterowników z serwerem.

W celu ułatwienia korzystania z technologii DDE, wprowadzono pojęcie biblioteki DDEML. Proces korzystający z tej biblioteki wywołuje funkcje zamiast wysyłać komunikaty i odbiera transakcje zamiast komunikatów.

Sieć LonWorks jest przystosowana do przesyłania informacji za pośrednictwem różnych środków transmisji, np.: sieci telefonicznych, fal radiowych, światłowodów, itp. Jedną z możliwości jest przesyłanie informacji parą skręconych przewodów – ten rodzaj środka transmisji został zastosowany do łączenia sterowników serii TA Xenta w sieci LonWorks.

Komunikacja modemowa w systemie dotyczy rozwiązań z wykorzystaniem łączy dzierżawionych, łączy komutowanych, łącza ISDN, łącza bezpośredniego. Przykładowy schemat podłączeń dla łącza komutowanego przedstawiono na rys. 4 [2].



Rys. 4. Schemat podłączeń dla łącza komutowanego (połączenie modemowe TAC Xenta 300/400)

W opisywanym systemie telemetrycznym występują ponadto dwa rodzaje połączeń bezpośrednich: z zastosowaniem karty PCLTA oraz przy pomocy modułu TA5702.

6. KONKLUZJE

Dzięki automatyzacji systemów ciepłowniczych uzyskuje się zmniejszenie zużycia energii cieplnej, poprawę stabilizacji temperatury ogrzewanych pomieszczeń oraz ciepłej wody użytkowej, ograniczenie pracochłonności obsługi podstawowej w węzłach cieplnych.

Do osiągnięcia tych celów konieczne jest w czasie eksploatacji zadanie odpowiednich dla danego obiektu nastaw, zapewnienie sprawności technicznej zastosowanych urządzeń, kontrola działania urządzeń, posiadanie informacji o stanach awaryjnych. Wybór rozwiązania technicznego podsystemów automatyki i kontroli zależny jest od zastanych w systemie układów technologicznych. Przy zastosowaniu urządzeń TAC Xenta 911 oraz 511 możliwe jest podłączenie nadzorowanych węzłów z serwerem w trybie *on-line* z wykorzystaniem sieci internetowej.

BIBLIOGRAFIA

[1] Dokumentacja produktów firmy TAC.
[2] Dokumentacja firmowa PI-W MERITUM.

DATA ACQUISITION AND PROCESSING IN THE TELEMONITORING SYSTEM OF HEAT TRANSMITTING NETWORK

Summary

Functional requirements for the system of automatics and remote supervision of an extensive heat network have been described. Tasks of the software for the communication and functional modules and the method of the system data base creation and management are presented. In communication for the remote control purposes the LonWorks standard with LonTalk protocol has bees applied. In the paper the representative nodes of the telemetric network and the types of communication. The system presented in the paper implemented for needs of PEC Toruń.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Lech Hasse[1], Józef Piasecki[2], Ludwik Spiralski[1]

[1]Katedra Metrologii i Systemów Elektronicznych, Politechnika Gdańska
[2]Instytut Łączności, Pracownia P-8 w Gdańsku

ZASTOSOWANIE OPROGRAMOWANIA COMSIS DO OCENY DYNAMICZNEJ ODPORNOŚCI NA ZAKŁÓCENIA CYFROWYCH UKŁADÓW KOMBINACYJNYCH

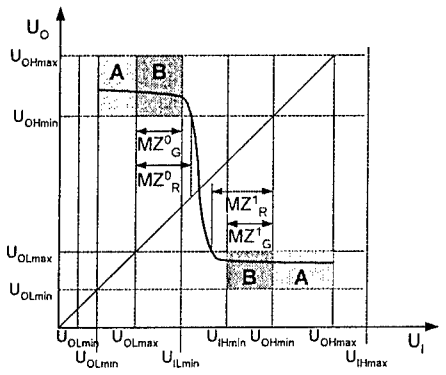
Streszczenie

Wyniki badania odporności układów cyfrowych na zakłócenie dynamiczne w istotny sposób zależą od parametrów sygnału stymulującego. Proponowana metodyka zakłada pobudzanie badanego układu sygnałem szumu białego o określonej szerokości pasma częstotliwości. Badania symulacyjne z modelem układu progowego oraz szybkiego komparatora z histerezą przeprowadzono w środowisku COMSIS. Uzyskane wyniki potwierdzają, że im węższe pasmo szumu pobudzającego badany układ, tym wyższy poziom widmowy mocy szumów jest wymagany do przełączenia stanu logicznego.

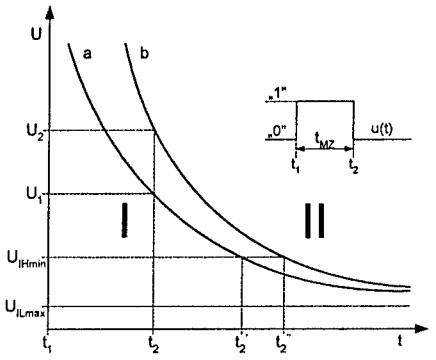
1. WPROWADZENIE

Obserwowana tendencja do zwiększania szybkości transmisji (zmniejszania czasów propagacji) cyfrowych układów i systemów, wymaga zmniejszania wymiarów fizycznych i mocy zasilania tych układów, przede wszystkim ze względu na konieczność nie przekraczania dopuszczalnych granicznych gęstości wydzielanej mocy. Konsekwencją tego jest coraz mniejsza różnica między mocą średnią sygnałów użytecznych oraz zawsze towarzyszących im sygnałów niepożądanych (zakłóceń). Rozrzuty produkcyjne parametrów oraz możliwości pracy układów w różnych warunkach (określanych przez dopuszczalne przedziały wartości napięć zasilających, obciążeń, temperatury, itp.) powodują konieczność wyznaczania tolerancji poziomów stanów logicznych, co pozwala z kolei na wyznaczenie statycznych (gwarantowanych i rzeczywistych) marginesów odporności na zakłócenia dla stanu wysokiego MZ^1_G i MZ^1_R oraz dla stanu niskiego MZ^0_G i MZ^0_R, odpowiednio, na podstawie poziomów przedstawionych na rys. 1 na charakterystyce napięcia wyjściowego U_o w funkcji napięcia wejściowego U_i. Obszary A i B wyznaczają strefy operacyjne dla poszczególnych stanów logicznych, przy czym A to strefy napięć w stanie ustalonym, zaś jako B oznaczono strefy, w których może wystąpić zmiana stanu logicznego na skutek występowania zakłóceń o poziomach dopuszczalnych dla obszarów A [1].

Zakłócenia o szerokim widmie (zwykle impulsowe), przenikające do układu cyfrowego przez linię transmisji, zasilania lub uziemienia, mogą powodować zmianę stanów logicznych prowadzącą do przekłamania i wadliwej pracy urządzeń czy systemów. Rzeczywista odporność układu na zakłócenia w warunkach dynamicznych określana bywa standardowo przez czas trwania $t_{MZ} = t_2 - t_1$ (rys. 2) prostokątnego impulsu zakłócającego o ustalonej amplitudzie powodującego chwilowe przekroczenie przez napięcie wyjściowe granicznego poziomu lub zmianę stanu logicznego. Dla ustalonego czasu trwania impulsu zakłócającego graniczna amplituda jest większa dla układów wolniejszych (krzywa a) o większych wartościach czasów propagacji ($U_2 > U_1$).



Rys. 1. Statyczne marginesy zakłóceń wyznaczane na charakterystyce przenoszenia przez poziomy ekstremalne oraz rzeczywiste napięć stanu wysokiego H i niskiego L ($MZ^0_G = U_{ILmax} - U_{OLMAX}$, $MZ^1_G = U_{OHmin} - U_{ILmin}$)



Rys. 2. Dynamiczne marginesy zakłóceń jako graniczne wartości amplitudy i czasu trwania impulsów (a – układy szybkie, b – układy wolne, I – obszar wartości dopuszczalnych, II – obszar wartości niedopuszczalnych)

Przy doborze testowych sygnałów stymulujących należy brać pod uwagę zarówno poziom, czas trwania, kształt przebiegów, jak i ich moc (energię).

2. OPIS METODY BADAŃ

Ponieważ oddziaływanie sygnałów zakłócających, zewnętrznych jak i pochodzenia wewnątrzukładowego, ma charakter przypadkowy, opracowana w Katedrze Aparatury Pomiarowej metodyka badania odporności na zakłócenia dynamiczne zakłada pobudzanie badanego układu sygnałem szumu białego (o stałej gęstości widmowej mocy i rozkładzie gaussowskim wartości chwilowych), co odpowiada rzeczywistemu charakterowi występujących zakłóceń [2]. Dzięki temu uzyskuje się bardziej adekwatną do warunków rzeczywistych ocenę odporności badanego układu na zakłócenia dynamiczne.

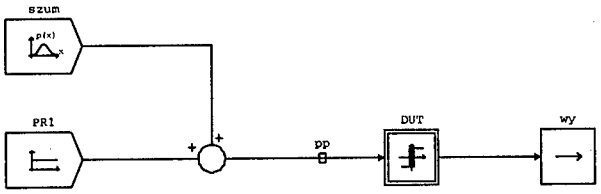
Szerokość pasma szumów powinna być zadana i zawierać częstotliwości większe od częstotliwości progowej badanego układu. Wymaga to zastosowania generatora szumu białego o dużej szerokości pasma i odpowiednim poziomie gęstości widmowej mocy. W przypadku trudności z zastosowaniem takiego generatora można zastosować sygnał stymulujący o węższym paśmie energetycznym, którego największa częstotliwość jest mniejsza od częstotliwości progowej badanego układu, a w paśmie tym zawarta jest

częstotliwość odpowiadająca odwrotności czasu propagacji badanego układu [3]. Poziom stymulującego sygnału szumowego zwiększa się aż do chwili zmiany stanu logicznego badanego układu, a iloczyn zadanej szerokości pasma energetycznego szumu i poziomu sygnału szumu wyrażonego w jednostkach gęstości widmowej mocy, odpowiadających tej zmianie stanu logicznego, przyjmuje się jako miarę odporności badanego układu na zakłócenia dynamiczne.

Modyfikacja tej metody polega na testowaniu badanego układu również sygnałem dynamicznym szumu o stałej gęstości widmowej mocy i rozkładzie gaussowskim wartości chwilowych, przy czym pasmo energetyczne sygnału szumowego podlega teraz ograniczeniom. Szerokość pasma energetycznego szumu jest zadana, a największa częstotliwość tego pasma jest mniejsza od częstotliwości progowej badanego układu i w paśmie tym korzystnie jest zawrzeć częstotliwość odpowiadająca odwrotności czasu propagacji badanego układu [3].

3. WYNIKI BADAŃ SYMULACYJNYCH

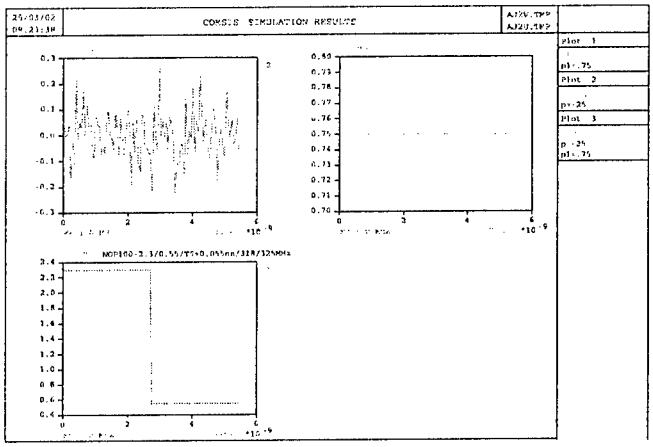
Efektywność nowej metody weryfikowano za pomocą badań symulacyjnych, korzystając z oprogramowania COMSIS firmy IPSIS. Zastosowano dwa modele symulacyjne badanego układu cyfrowego DUT: model LV (*Low Voltage*) detektora progowego oraz model szybkiego komparatora z histerezą. Na rys. 3 przedstawiono model symulacyjny DUT, zbudowany przy zastosowaniu detektora progowego oraz szeroko-pasmowego źródła szumu białego. Składowa stała napięcia, dodawana w sumatorze z szumem białym, służy do ustawienia napięcia o zbliżonej wartości (wyznaczona różnica wynosi 0.2 V powyżej lub poniżej katalogowych progów napięciowych) do poziomów detekcji (niskiego L lub wysokiego H) dla badanego układu DUT, zalecanych przez producentów w celu zapewnienia tym układom stabilnych warunków pracy podczas zmiany stanów logicznych.



Rys. 3. Schemat blokowy modelu symulacyjnego badania dynamicznej odporności na zakłócenia układów cyfrowych: szum – generator szumu białego, PR1 – napięcie stałe, pp – sygnał wejściowy (szum + PR1), DUT – badany układ cyfrowy, wy – sygnał wyjściowy.

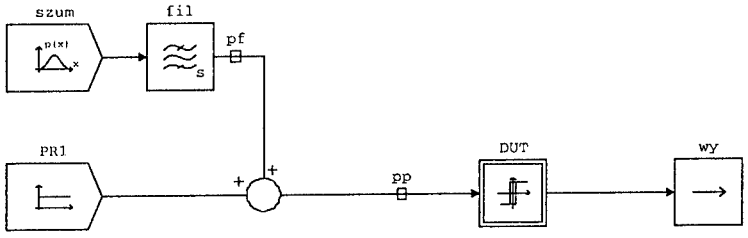
Przykładowe wyniki symulacji dla układu cyfrowego SN74LVC06 firmy Texas Instrument pokazano jako ekspozycje graficzne z programu COMSIS na rys. 4. Symulacje przeprowadzono przy następujących wartościach parametrów: szerokość pasma generatora szumu białego – 325 MHz (częstotliwości graniczna badanego układu wynosi 318 MHz), stała PR1 = 0.75 V dla zmiany stanu logicznego ze stanu H do stanu L, zaś dla zmiany z poziomu L do H – 2.1 V, zgodnie ze specyfikacją firmową: poziom L – 0.55 V, poziom H – 2.3 V (zarówno dla wejścia, jak i wyjścia DUT). Dla obu przypadków przyjęto jednakową liczbę przetwarzanych punktów charakterystycznych (NOP) równą 100 oraz krok pomiarowy (TS) wynoszący 0.055 ns. Stwierdzono, że przełączenie badanego układu

z poziomu H do L następuje przy pobudzeniu szumem o poziomie mocy p = –25 dB, zaś z poziomu niskiego L do H poziom mocy szumu niezbędny do zmiany stanu logicznego wynosi p = –26 dB.



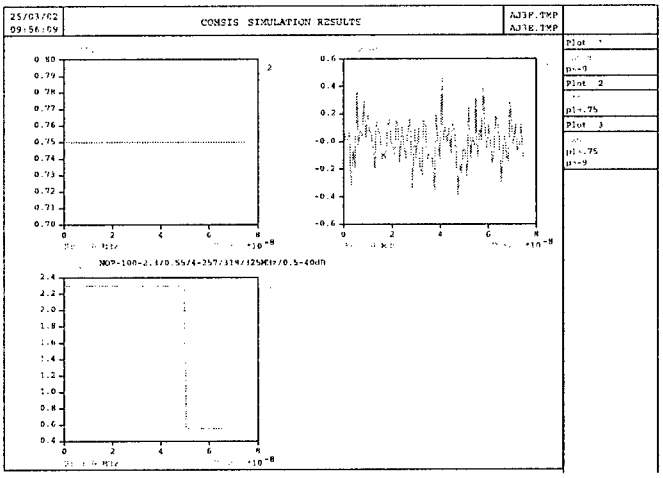
Rys. 4. Ilustracja badań symulacyjnych przy przełączeniu DUT z poziomu H do L i pobudzeniu jego wejścia szumem białym o poziomie mocy 25 dB (stała PR1 = 0.75 V).

Powyższe wyniki porównano z wynikami symulacji przeprowadzonych z wykorzystaniem modelu przedstawionego na rys. 5. Umożliwia on weryfikację zmodyfikowanej metody pomiaru odporności układów cyfrowych przy zadanej szerokości pasma energetycznego zawierającej częstotliwość odpowiadającą odwrotności czasu propagacji badanego układu i o największej częstotliwości zawartej w tym paśmie mniejszej od częstotliwości progowej badanego układu. Ograniczenie szerokości pasma sygnału szumowego uzyskano stosując odpowiedni filtr pasmowy, pozwalający na określenie szerokości pasma sygnału stymulującego, tłumienia poza pasmem przenoszenia oraz typu aproksymacji (filtr Bessela, Butterwortha, Czebyszewa, Cauera).



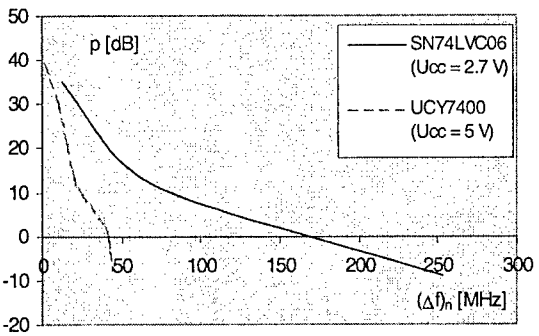
Rys. 5. Schemat blokowy modelu symulacyjnego badania dynamicznej odporności na zakłócenia układów cyfrowych przy zadanym paśmie szumu: szum gauss. – generator szumu białego, fil – filtr pasmowy analogowy, pf – sygnał na wyjściu filtru, PR1 – stała napięcia w [V], pp – sygnał wejściowy (szum gauss.+PR1), DUT – badany układ cyfrowy, WY – sygnał wyjściowy.

W celu uzyskania wyników porównawczych, badania symulacyjne za pomocą modelu z rys. 5 przeprowadzono również dla układu SN74LVC06, którego katalogowy czas propagacji wynosi 3.9 ns, a odpowiadająca mu częstotliwość wynosi 256.4 MHz i jest mniejsza od częstotliwości granicznej 318 MHz. We wszystkich symulacjach przyjęto górną częstotliwość pasma przenoszenia równą 257 MHz, natomiast zmianie podlegała częstotliwość dolna filtru (przyjęto typ aproksymacji Cauera, przy której uzyskuje się maksymalnie strome zbocza filtru oraz wysoką efektywność, zarówno przenoszenia, jak i tłumienia; parametry tłumienia filtru: wartość dolna – 0.5 dB , wartość górna – 40 dB).



Rys. 6. Ilustracja badań symulacyjnych przy przełączaniu DUT z poziomu H do poziomu L i pobudzeniu wejścia pp szumem o paśmie od 4 do 257 MHz i poziomie mocy p = –9 dB.

Na rys. 6 pokazano przykładowe wyniki badań modelu z rys. 5 dla pasma szumu podawanego na wejście pp układu DUT od 4 do 257 MHz. Zależność poziomu gęstości widmowej mocy szumów niezbędnej do zmiany stanu logicznego dla wolnych układów logicznych (UCY7400) oraz szybszych układów CMOS (SN74LVCO6). DUT od szerokości pasma sygnału stymulującego przedstawiono na rys. 7.



Rys. 7. Wymagany poziom gęstości widmowej mocy p(f) niezbędny do zmiany stanu logicznego DUT w funkcji szerokości pasma szumu przy opisanych badaniach symulacyjnych.

Przeprowadzone badania symulacyjne potwierdzają możliwość określania dynamicznej odporności na zakłócenia układów cyfrowych według założeń przyjętych w zgłoszeniach patentowych [3].

6. KONKLUZJE

Do przełączenia układu cyfrowego potrzebna jest określona energia dostarczana do badanego układu przez sygnał pobudzający. Przeprowadzone badania symulacyjne potwierdzają celowość określania dynamicznej odporności na zakłócenia układów cyfrowych według założeń przyjętych w zgłoszeniach patentowych [3]. Uzyskane wyniki symulacji potwierdzają tezę, że im węższe jest zadane pasmo pobudzającego sygnału szumowego, tym wyższy poziom mocy widmowej jest niezbędny do zakłócającej zmiany stanu logicznego badanego układu. Tym samym uzasadniono efektywność badania dynamicznej odporności na zakłócenia za pomocą sygnału szumowego o zadanym paśmie. Wyniki symulacji zostały potwierdzone w przeprowadzonych pomiarach dynamicznej odporności na zakłócenia według proponowanej metody.

BIBLIOGRAFIA

[1] Kołodziejski J.F., Spiralski L., Stolarski E.: *Pomiary przyrządów półprzewodnikowych.* WKiŁ, Warszawa 1990.

[2] Piasecki J., Spiralski L., Hasse L.: *Pomiar dynamicznej odporności na zakłócenia układów cyfrowych z zastosowaniem wzorcowego sygnału szumowego.* Krajowy Kongres Metrologii KKM'2001, Warszawa 24-27.06.2001, t. II, Sesja 7, s. 313-316.

[3] Piasecki J., Spiralski L., Hasse L.: *Sposób badania odporności na zakłócenia dynamiczne układów cyfrowych.* Zgłoszenia patentowe: P-344507 z dnia 06.12.2000 oraz P-352567 z dnia 01.03.2002.

EVALUATION OF LOGIC CIRCIUT DYNAMIC NOISE IMMUNITY USING COMSIS SOFWARE

Summary

Results of noise susceptibility measurement in digital circuits depend hardly on stimulus signal parameters. The method of dynamic noise immunity measurement and simulation using a white noise signal having assumed bandwidth has been presented. The simulation models containing a threshold circuit and a fast comparator with hysteresis loop were prepared in COMSIS environment. The achieved results confirm that the narrower bandwidth of noise stimulus the higher level of noise power spectrum is necessary to switch the digital circuit under test.

ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 2 Seria: Technologie Informacyjne 2004

Henryk Lasota

Katedra Systemów Elektroniki Morskiej, Politechnika Gdańska

KSZTAŁCENIE INŻYNIERÓW ICT W ŚWIECIE PERMANENTNYCH ZMIAN

Streszczenie

Wyzwania, jakie stawia przed światem rozwój technologiczny, są jakościowo nowe. Zjawisko, zwane „budowaniem społeczeństwa informacyjnego", skłania środowiska akademickie i zawodowe do pogłębionej refleksji, zaś przedsiębiorców, zwłaszcza w trudnej branży szeroko pojętej telekomunikacji, zmusza do złożonych, ryzykownych, przedsięwzięć.
Dyskusja problemów kształcenia inżynierów dla potrzeb przedsiębiorstw działających w obszarze ICT, przewidziana w ramach seminarium „Telekomunikacja – świat permanentnych zmian" organizowanego przez stowarzyszenia Inżynierów Telekomunikacji (SIT) i Elektryków Polskich (SEP), ma na celu zgromadzenie doświadczeń wymienionych środowisk, z myślą o zbudowaniu na Wydziale ETI PG systemu zarządzania programami studiów, reagującego bez zbędnych opóźnień na permanentne zmiany otoczenia.

1. WSTĘP

W ostatnim piętnastoleciu, a nawet pięcioleciu, doszło w Kraju i na świecie do poważnych zmian w technice, technologii i otoczeniu rynkowym, zwłaszcza w dziedzinach zwanych „high-tec". Zjawisko dynamizacji zmian przybrało charakter trwały. Inaczej niż w poprzednich epokach, nie występują stany quasi-ustalone. Trzeba przyjąć do wiadomości paradoks współczesności, że jedynym stałym jej elementem jest ciągła zmienność.

Gwałtownie poszerza się obszar wiedzy specjalistycznej. Nie jest możliwe jej pełne opanowanie w toku studiów. Czas przeznaczony na studia stacjonarne jest stały, stąd konieczne są zmiany zakresu merytorycznego i metodycznego. Nie należy jednak ulegać pokusie zawężania specjalizacji. Przeciwnie, za uzasadnione należy uznać próby integracji różnych obszarów działalności dydaktycznej naszego Wydziału, kształcącego elektroników, telekomunikantów, automatyków i informatyków, wokół wspólnego określenia – Technologie Informacyjne (IT) lub, ściślej, Techniki Informacyjno – Komunikacyjne (ICT).

Dla sprostania wyzwaniom teraźniejszości, należy nie tyle aktualizować programy studiów, co je przebudować, uwzględniając zmiany, do jakich doszło zarówno w otoczeniu technologicznym, jak i na rynku zatrudnienia. Wydział Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej (WETI PG) podjął wyzwanie opracowania systemu

budowy programów studiów na miarę potrzeb otoczenia, który korzysta z rosnących możliwości środków informatycznych, w tym systemów inżynierii wiedzy [1].

2. WYMAGANIA CYWILIZACYJNE

Współczesna cywilizacja jest światem permanentnych zmian. Gwałtowny rozwój technologiczny i stale wzrastające zapotrzebowanie rynkowe są dwoma sprzężonymi motorami wzajemnie napędzającego się mechanizmu społeczno-gospodarczego. Stale ulepszana technologia i rosnące wymagania klientów powodują rosnącą złożoność problemów, jakie rozwiązywać muszą przedsiębiorcy, zwłaszcza ci, którzy są dostawcami towarów i usług związanych z wysokimi technologiami. Nadzieją firm na osiągnięcie bądź utrzymanie zadowalającej pozycji na tym burzliwym rynku są m.in. kadry wykształcone w uczelniach technicznych, na wydziałach prowadzących kierunki związane z obszarem ICT, czyli inżynierowie elektroniki, telekomunikacji, automatyki i robotyki, informatyki.

W rezultacie zmian wynikających w znacznej części z rozwoju technologicznego, następują zmiany cywilizacyjne określane hasłem „Społeczeństwo Informacyjne". Towarzyszy temu znacząca, przyspieszona ewolucja potrzeb rynku pracy i oczekiwań społecznych w odniesieniu do absolwentów uczelni technicznych. Liczy się, mianowicie, nie tyle wiedza, co umiejętność jej twórczego przetworzenia – dla zaspokajania pojawiających się potrzeb, a nawet – dla kreowania potrzeb i, tym samym, dla budowania dla swojej firmy (choćby skromnych) nisz rynkowych.

Na uniwersytetach technicznych i na środowiskach zawodowych spoczywa obowiązek podjęcia pilnych i skutecznych działań zwiększających szanse młodzieży opuszczającej mury politechnik, na sprostanie wymaganiom współczesności, na pełnienie roli elity kadrowej w przedsiębiorstwach. Po absolwencie Wydziału ETI PG rynek oczekuje dobrej jakości. Dla Wydziału (i dla jego absolwentów) byłoby najlepiej, gdyby jakość ta była lepsza od spodziewanej, a z pewnością zdecydowanie lepsza niż obecnie. Warto więc, i należy, zbudować system ciągłej przebudowy programów, „rozpięty" między ewoluującym otoczeniem technologicznym, fluktuującym otoczeniem ekonomicznym i chaotycznie zmieniającym się rynkiem pracy, system umożliwiający modyfikowanie elementów oferty dydaktycznej Wydziału, w miarę pojawiających się potrzeb.

Takie postawienie problemu jakości kształcenia sugeruje systemowe, projektowe podejście do jego rozwiązania. Pierwszym etapem takiego projektowania jest określenie perspektywicznych celów i zadań procesu dydaktycznego, prowadzące w konsekwencji do decyzji odnośnie modelu studiów, który ma być realizowany na Wydziale, oraz do zdefiniowania sylwetki absolwenta każdego z kierunków studiów. Wspólny zestaw pożądanych umiejętności, jaki został przyjęty dla wszystkich kierunków, pozwala określić w miarę jednorodny profil absolwenta Wydziału ETI.

3. CELE PROCESU KSZTAŁCENIA

Senat naszej Uczelni zadeklarował, że *„Misją Politechniki Gdańskiej jest kształcenie kadr inżynierskich, menedżerskich i naukowych (...)"* [2]. W praktyce oznacza to edukację[1] wybranej części młodzieży, należącej do elity wśród rówieśników pod względem

[1] edukacja (łac. *educatio* – wychowanie, od *ducerere* – prowadzić; *dux, ducis* – przywódca) – wychowywanie, kształcenie pod odpowiedzialnym przywództwem

możliwości intelektualnych, motywacji do nauki i skuteczności przyswajania wiedzy. Elitarność ta jest szczególnie wyraźna w przypadku tak złożonej dziedziny, jak ICT.

3.1. Cel formalny

Za cel formalny kształcenia na uczelni technicznej należy uznać nadanie absolwentowi tytułu zawodowego, przewidzianego w Zarządzeniu MEN z 9.04.1992. W przypadku politechnik w grę wchodzą następujące tytuły:

- inżynier – dla absolwenta „wyższych zawodowych studiów technicznych" lub
- magister inżynier – dla absolwenta „magisterskich wyższych studiów technicznych".

Formalnie nie ma wymagania, by mgr inż. miał kwalifikacje inżyniera. Natomiast zwyczajowo – takie są oczekiwania otoczenia, zarówno zawodowego jak i społecznego.

3.2. Cel dydaktyczny

Celem dydaktycznym jest rozwijanie u studentów wiedzy i umiejętności zawodowych, z pożądanym poziomem kompetencji – możliwym do osiągnięcia w młodym wieku. Sprzyja temu, zgodnie z zaleceniami metodyków kształcenia, zachowanie odpowiednich proporcji między równoczesnym zdobywaniem wiedzy teoretycznej, technologicznej i umiejętności praktycznych.

Wyborem o znaczeniu strategicznym dla wizerunku Wydziału i dla umiejętności zawodowych jego absolwentów, jest rozstrzygnięcie w budowanym programie studiów formalnie nie istniejącego problemu zdobywania kwalifikacji inżynierskich na studiach magisterskich. Wydział liczy na wsparcie środowisk zawodowych i przedsiębiorców przy rozwiązywaniu tego problemu.

3.3. Cele pedagogiczne

Cele pedagogiczne kształcenia, choć wymienione na ostatnim miejscu, są w istocie najważniejsze, jako że koniec studiów jest zwieńczeniem okresu dojrzewania. Absolwent ma być człowiekiem, który dorósł do „twórczej i odpowiedzialnej pracy Inżyniera"[2].

Dla kadry dydaktycznej Wydziału oznacza to współuczestnictwo w kształtowaniu osobowości studenta, co przekłada się na współodpowiedzialność za właściwy rozwój duchowy młodego człowieka.

3.3.1. Rozwój intelektualny

Rozwój umysłowy to nie tylko poznawanie wiedzy i zdobywanie umiejętności zawodowych na odpowiednim poziomie, lecz także rozumienie otaczającego świata w całej jego złożoności. Składa się na to między innymi:

- poznanie i rozumienie nauki – jej możliwości i ograniczeń, oraz
- poznanie i zrozumienie świata współczesnych technologii – ich możliwości i niebezpieczeństw.

Bowiem „tylko ten, kto rozumie rzeczywistość, potrafi się w niej odnaleźć, zmieniać ją i odnosić w niej sukcesy".

[2] cytat ze ślubowania (vide indeks studencki Politechniki Gdańskiej)

3.3.2. Kształtowanie osobowości

Jednym z sześciu podstawowych składników poczucia własnej wartości jest kompetencja rozumiana jako świadomość, że umiemy coś zrobić [3]. Jest ona szczególnie istotna w warunkach współczesnej złożoności technologicznej i społecznej, w jakiej przyjdzie działać absolwentowi studiów technicznych. Dla wyrobienia uzasadnionego poczucia kompetencji, zalecana jest dbałość o:

1. rozsądne proporcje między ilością wymaganej pracy a szansą na sukces (zaliczenie), oraz
2. równowagę między pasywnym i aktywnym uczestnictwem w procesie kształcenia, między aspektami teoretycznymi i praktycznymi, między indywidualnym słuchaniem wykładów a zespołowym poszukiwaniem rozwiązań w grupie projektowej.

W pierwszym przypadku chodzi o równomierne rozłożenie wymagań w stosunku do możliwości percepcyjnych w toku całych studiów. Oznacza to, z jednej strony, potrzebę unikania na pierwszych semestrach przeciążenia, które powodowane jest przez dużą liczbę wykładów o sporym ładunku wiedzy teoretycznej na zaawansowanym poziomie, nie dającej się dobrze przyswoić w wymaganym czasie. W praktyce przeciążenie to prowadzi do licznych niepowodzeń i jest przyczyną „odsiewu", dotykającego także studentów wystarczająco zdolnych, lecz nie dość jeszcze odpornych psychicznie. Z drugiej strony, oznacza to wzbogacenie edukacji na wyższych latach zarówno o zajęcia projektowe, umożliwiające przekształcenie, przez dojrzalszych już studentów, wcześniej zdobytej wiedzy w umiejętności zawodowe, jak i o wykłady uzupełniające z przedmiotów podstawowych związanych z tematyką dyplomową, prowadzone na zaawansowanym poziomie teoretycznym i/lub praktycznym.

W drugim przypadku, chodzi m.in. o to, że wiedza teoretyczna, obecnie przekazywana na niższych latach bez wyraźnego, z braku czasu, wskazania na związek z praktyką, jest odbierana jako abstrakcja. Z kolei, brak umiejętności zastosowania tak potężnego ładunku wiedzy rodzi w efekcie frustrację i wytracanie młodzieńczego dynamizmu. Stoi to w sprzeczności z apostolskim nauczaniem, by „nie gasić ducha".

3.3.3. Rozwój społeczny

Do społecznych cech absolwenta poszukiwanych na rynku pracy należą:

- umiejętność pracy zespołowej (współpraca w przedsiębiorstwie i na zewnątrz),
- umiejętność skutecznej komunikacji z otoczeniem *sensu largo* (pozyskiwanie informacji o potrzebach, w tym nie uświadomionych przez potencjalnych klientów, przekazywanie informacji o możliwościach firmy).

Dysponowanie przez absolwenta umiejętnościami tego typu oznacza właściwe jego przygotowanie do pełnienia oczekiwanych ról społecznych: zarówno roli odpowiedzialnego pracownika firmy zatrudniającej, jak i (w dłuższej perspektywie, po zdobyciu odpowiedniego doświadczenia życiowego i zawodowego) samodzielnego przedsiębiorcy zapewniającego zatrudnienie innym, w tym kolegom.

4. BUDOWANIE I PIELĘGNACJA PROGRAMÓW STUDIÓW

O ile cele edukacyjne omówione w poprzednim rozdziale są oczywiste, o tyle ich osiągnięcie nie jest łatwe. Wymagają one, bowiem, przełożenia na konkretne zagadnienia i pojedyncze działania, nierzadko wzajemnie wykluczające się.

Dobór treści programowych oraz wybór metodyki przekazywania wiedzy studentom, to zagadnienia klasyczne, jednakże stały rozwój nowych technik i technologii oznacza sytuację jakościowo nową. Opracowanie systemu sprawnej i skutecznej aktualizacji programów studiów, umożliwia uruchomienie mechanizmu ciągłej poprawy i czyni możliwym stałe utrzymywanie pożądanego poziomu jakości edukacji. Jest to zagadnienie kluczowe dla utrzymania przez Wydział przewagi konkurencyjnej na rynku edukacyjnym.

4.1. Metoda tradycyjna [4]

Tradycyjna strategia tworzenia zmodyfikowanych programów studiów zasadza się na podejściu indywidualnym, w którym decyzje co do treści programowych i sposobów ich przekazywania, pozostawia się osobom prowadzącym zajęcia. W programach i planach nauczania operuje się nazwami przedmiotów, które są jednostkami n x 15 – godzinnymi, trwającymi całkowitą liczbę semestrów. Negocjowanie planów zajęć (tzw. siatek godzinowych) i zmian w planach odbywa się pomiędzy kierownikami Katedr. Z kolei negocjowanie programów zajęć, i ewentualnych zmian w treściach programowych, odbywa się pomiędzy zainteresowanymi, tzn. wykładowcami odpowiedzialnymi za zajęcia, w których występują te same, lub podobne, zagadnienia.

Budowa programu odbywa się tu zasadniczo metodą wstępującą, od szczegółów do ogółu (*bottom-up*). Taka metodyka modyfikacji stosowana jest powszechnie, nie tylko na naszym Wydziale, ale na tzw. „całym świecie", zarówno w pracy bieżącej, jak i przy okazji oficjalnie deklarowanych akcji zmiany programów. Ma ona istotne ograniczenia, gdyż:

- reprezentowany jest przede wszystkim punkt widzenia wykładowców i Katedr; zaś
- koordynacja zawartości merytorycznej i metodyki odbywa się z perspektywy przedmiotów, traktowanych *de facto* jak quasi-niezmienniki i ogranicza się do negocjacji pomiędzy wykładowcami dotyczących powiązań między przedmiotami oraz adaptacyjnych, jedynie, korekt indywidualnych przedmiotów.

Jak uczy doświadczenie wszystkich stabilnych instytucji edukacyjnych, rezultatem takich prac koordynacyjnych jest najczęściej brak istotnej, całościowej poprawy programów. Okazuje się, mianowicie, że łatwo jest osiągnąć zmiany kosmetyczne, natomiast trudno jest doprowadzić do istotnych zmian merytorycznych i metodycznych. Nade wszystko, zaś, **nie udają się zmiany systemowe!** W efekcie nie można budować programów perspektywicznie, w sposób uwzględniający ciągłe zmiany.

Z punktu widzenia systemowego, takie rezultaty, chociaż **niepożądane**, są całkiem **naturalne**. Przyczyną niemożności osiągnięcia spodziewanego rezultatu jest fakt, że podejście tradycyjne nie uwzględnia, w istocie, realiów otaczającego świata. Zakłada ono:

- dobrą wolę wszystkich udziałowców (słusznie),
- niesprzeczność interesów katedr (niesłusznie), a przede wszystkim
- pożyteczność indywidualnie negocjowanych kompromisów.

O ile pierwsze założenie można uznać za uzasadnione, a drugie za możliwe do osiągnięcia (choć jedynie w szczególnie sprzyjających okolicznościach), o tyle zasadność ostatniego założenia jest wątpliwa. Więcej, przy braku podejścia systemowego takie kompromisy mogą być – i często są - sprzeczne z deklarowanymi celami.

W podejściu tradycyjnym, wykorzystującym intuicyjne wyczucie kadry dydaktycznej, zazwyczaj nie są formalnie wyartykułowane cele, sposoby ich osiągania ani strategia wdrożenia zmian. Nie są ponadto identyfikowani wszyscy tzw. udziałowcy / interesariusze, czyli osoby, organizacje i czynniki zaangażowane w działania lub przez te działania dotykane. Nie uwzględnia się więc w sposób jawny różnorakich zagrożeń dla procesu

zmian. Powoduje to z kolei brak skutecznego przeciwdziałania niekorzystnym wpływom i niepożądanym obrotom spraw.

I na koniec – *last but not least* - towarzyszy całej działalności modyfikacji programów **niedostatek udokumentowania**, powodujący brak przejrzystości, a w konsekwencji poważne utrudniający weryfikację rezultatu.

4.2. Metoda systemowa [4]

Podejście systemowe polega na potraktowaniu budowy programów studiów jako przedsięwzięcia, którego realizacja wymaga zastosowania metodyki zarządzania złożonymi projektami. Postępowanie przy projektowaniu treści programowych i metodyki nauczania ma tu charakter zstępujący – od ogółu do szczegółów (*top-down*), a wszystkie etapy i fazy działania są dokumentowane i weryfikowane. Rezultatem działania jest program studiów w postaci umożliwiającej wykorzystanie środków i systemów informatycznych.

Inżynieria projektowania programów studiów wymaga nie tylko zaprojektowania zawartości i formy zajęć, ale także budowy mechanizmu zapewnienia jakości uwzględniającego konieczność realizacji ciągłej zmiany ewolucyjnej. Proces projektowania i modyfikacji programów musi być odpowiednio zarządzany. Należy jasno określić procedury wypracowywania i podejmowania decyzji, harmonogramy dochodzenia do poszczególnych celów, sposoby włączania do działań poszczególnych udziałowców, zwłaszcza wykładowców i studentów.

Istotne jest udokumentowanie systemu edukacyjnego Wydziału. Dokumentacja jest bowiem niezastąpionym nośnikiem komunikacji między udziałowcami (prowadzący zajęcia między sobą – teoretycy/praktycy, studenci, klienci), zarówno w procesach negocjacji treści i metodyki, jak i w samych procesach nauczania (wykładowcy) i w procesie uczenia się (studenci). Wykonanie **dokumentacji** wymaga zainwestowania sporego czasu i wysiłku, jednak w zamian Wydział uzyskuje **wydajne narzędzie zarządzania dydaktyką**, które umożliwia bieżącą obserwację procesu dydaktycznego i stałą pielęgnację programów studiów (utrzymanie na pożądanym poziomie aktualności i skuteczności).

Podejście systemowo-projektowe przewiduje następujące kluczowe elementy działania:

- specyfikacja celów nauczania (sformułowanie ogólnych celów w postaci np. sylwetki absolwenta określającej cechy „dzieła" opuszczającego Wydział),
- określenie tzw. wymagań funkcjonalnych (wybór modelu/modeli pozyskiwania wiedzy i umiejętności; decyzje co do kolejności),
- określenie edukacyjnych celów „operacyjnych" (wybór zagadnień i form kształcenia; określenie wymaganych poziomów wiedzy i/lub umiejętności – określenie rodzaju zajęć) i ich wariantowa analiza,
- specyfikacja zawartości przedmiotów (projektowanie konkretnych zajęć – wykłady, ćwiczenia, laboratoria, projekty; rozwiązanie nieuniknionych konfliktów między ilością a jakością, wypracowanie właściwych kompromisów),
- wypracowanie polityki skutecznego wdrażania i pielęgnacji „produktu" jakim jest opracowywany program studiów.

Inaczej mówiąc: najpierw zdefiniowanie celów, potem budowa modelu, ustalenie kryteriów zgodności z celami, projekty przedmiotów, a na koniec – wypracowanie mechanizmu stałego doskonalenia.

Wspomniany wyżej problem poziomów wiedzy / umiejętności w poszczególnych zagadnieniach tematycznych, wiąże się bezpośrednio z problemem koniecznych kompromisów wynikających z ograniczeń czasowych. W zależności od wagi zagadnień,

mających charakter tzw. obiektów wiedzy, pożądane i możliwe do osiągnięcia poziomy wiedzy i/lub umiejętności zostały określone dla potrzeb edukacji na Wydziale w skali 5 – stopniowej. Trzy poziomy wiedzy A, B, C oznaczają, że: A – zostanie podana informacja o istnieniu danego zagadnienia i jego ogólna charakterystyka (lub, że będzie to podsumowanie zagadnienia znanego), B – zagadnienie zostanie przedstawione w sposób dokładny, C – zagadnienie zostanie przedstawione szczegółowo, tak by dać podstawę dla zdobycia umiejętności stosowania go w praktyce. Dwa poziomy umiejętności, to: D – umiejętność wykorzystania zagadnienia w sytuacjach typowych lub E - wykorzystania w złożonych sytuacjach problemowych.

4.3. Wybór treści programowych [1,5]

Podejście systemowe do budowania programu polega na zdefiniowaniu **zakresu wiedzy** (*body of knowledge*) – tego wszystkiego, co powinien opanować absolwent danego kierunku studiów i danej specjalności. Zakres wiedzy ma strukturę hierarchiczną, trójpoziomową. Poziom najwyższy to obszary wiedzy (areas), odpowiadające tematycznym partiom materiału do opanowania przez studenta danego kierunku w czasie całego okresu trwania studiów. Obszary dzielone są na mniejsze składniki – **jednostki tematyczne** (*units*), stanowiące środkowy poziom obowiązującego zakresu wiedzy. Te z kolei składają się z **zagadnień** (*topics*), będących najniższym poziomem zakresu wiedzy.

Istotą metody systemowej jest to, że **ani jednostki ani obszary nie stanowią przedmiotów**. Co więcej, zagadnienia poszczególnych jednostek tematycznych mogą, i powinny, być realizowane przez różne przedmioty rozłożone w całym programie studiów – od pierwszego do ostatniego roku, w miarę wzrostu kompetencji i zaawansowania wiedzy studenta. Podejście takie ma dwie zalety, umożliwiające realizację omówionych wcześniej celów dydaktycznych. Po pierwsze, można uniknąć przeładowania pierwszych lat studiów przedmiotami teoretycznymi, które mogą zniechęcać do dalszej pracy studentów lat niższych ze względu na brak odniesień do praktyki. Po drugie, możliwe staje się wprowadzanie materiału teoretycznego bezpośrednio przed tym, jak staje się on potrzebny w przedmiocie technologicznym. Programy poszczególnych przedmiotów mogą zatem obejmować zagadnienia z różnych jednostek, pokrywane w całości lub w części.

5. WSPÓŁPRACA MIĘDZYŚRODOWISKOWA

Umiejętność rynkowego wykorzystywania nowych technik i technologii jest, kluczowa dla przedsiębiorstw. Mogą one, bowiem, budować i wzmacniać swą pozycję rynkową przez opracowywanie i wdrażanie nowych produktów – towarów bądź usług.. Kwalifikacjami na wagę złota okazują się więc umiejętność dokonywania wnikliwej analizy potrzeb rynku oraz znajomość metodyki zarządzania przedsięwzięciami.

W obecnej sytuacji uczelnie, przedsiębiorcy i środowiska zawodowe stają się, w większym niż wcześniej stopniu, naturalnymi sprzymierzeńcami. Każda ze stron ma do wniesienia do gry swe doświadczenie, swoje umiejętności, swoje zasoby. Każda ze stron powinna też ujawnić swoje potrzeby, zgłosić problemy do rozwiązania.

Uczestnicy spoza Uczelni zostaną poproszeni o pełnienie roli Klientów reformowanego systemu edukacji na Wydziale i o pomoc w pozyskiwaniu wymagań na ten system. W szczególności interesujące będzie wskazanie, o jakie elementy wiedzy i umiejętności – zwłaszcza z obszarów pozatechnicznych, powinny w Ich opinii zostać uzupełnione programy studiów. Metoda systemowa, dając możliwość konstruowania zajęć „na miarę",

nawet w blokach kilkugodzinnych, pozwala uwzględnić każdy postulat zwiększający dopasowanie Absolwenta Wydziału do potrzeb Rynku. Dla osiągnięcia dopasowania wskazane byłoby także umożliwienie wszystkim studentom odbywania praktyk zawodowych w firmach z branży.

Reagując na potrzeby środowiska zawodowego Wydział ETI prowadzi różnego rodzaju studia podyplomowe, w tym studia w zakresie Zarządzania Przedsięwzięciami Telekomunikacyjnymi i Teleinformatycznymi, które są m.in. poligonem doświadczalnym przed uruchomieniem, w następnym roku akademickim, dwusemestralnego projektu grupowego dla wszystkich studentów Wydziału.

Debata organizowana przez Koło PG Stowarzyszenia Inżynierów Telekomunikacji (SIT) oraz Oddziały Gdańskie SIT i SEP, na specjalnej sesji niniejszej Konferencji, związanej ze Światowym Dniem Telekomunikacji 2004, poświęcona jest kształceniu inżynierów szeroko pojętej telekomunikacji, dla potrzeb przedsiębiorstw działających w trudnej branży ICT. Jej rezultatem ma być zdynamizowanie współpracy międzyśrodowiskowej w tym zakresie.

Konkretny efekt przyniosła już współpraca Koła SIT PG z Oddziałem Gdańskim SEP. Mianowicie, na Wydziale zostaną zorganizowane kursy umożliwiające chętnym studentom uzyskanie SEP-owskich uprawnień eksploatacyjnych w zakresie do 1kV. Niewątpliwie, zwiększy to ich szanse na rynku pracy.

BIBLIOGRAFIA

[1] Komisja Programów Studiów Wydziału ETI Politechniki Gdańskiej: *Raport KPS 2002*, Gdańsk, luty 2003 r.

[2] Uchwała Senatu Politechniki Gdańskiej nr 148/02/XIX z 27 marca 2002 r.

[3] B. B. Youngs: *The 6 Vital Ingredients of Self-Esteem, How To Develop Them In Your Students,*. W: G. Dryden, J. Vos: *Rewolucja w uczeniu*, Wydawnictwo Moderski i S-ka, Poznań 2000

[4] P. Bots, W. Thissen: *Negotiating Knowledge in Systems Engineering Curriculum Design: Shaping the Present While Struggling with the Past*, IEEE Trans. Systems, Man, And Cybernetics, vol.30 (2000), 2, 197-203.

[5]*Computing Curricula 2001 – Computer Science*, The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, Final Report, December 15, 2001.

FORMING ICT ENGINEERS IN THE WORLD OF PERMANENT CHANGE

Summary

Qualitatively new challenges have been created by technological development. The phenomenon being called "the construction of information society" induces university people and professionals to a deepened reflexion while forcing entrepreneurs to difficult and risky actions, particularly at the telecomm market.

Problems of formation of engineers for enterprises working in ICT branch will be discussed in the frame of a seminar organized by SIT and SEP (associations of telecomm and electrical engineers) on "Telecomm – the world of permanent change". The debate should gather experience for constructing at the ETI Faculty a system of curricula management, correctly reacting to permanent changes in the environment.

List of Contributors

Adamczyk Arnold, 433
Ambriola Vincenzo, 671
Antczak Bartłomiej, 645
Arató Péter, 37
Babiarz Patryk, 199
Bajor Michał, 207
Banaszczyk Anna, 207
Bartosiński Bogdan, 785
Bikonis Krzysztof, 883
Blok Marek, 115, 125
Bluemke Ilona, 611
Błaś Mariusz, 327
Bobek-Billewicz Barbara, 191
Bobkowska Anna, 619
Brudło Piotr, 45
Buchalski Zbigniew, 717
Buszkiewicz Paweł, 207
Byczkowska-Lipińska Liliana, 553
Cegielski Marcin, 553
Chlewicka Aldona, 891
Cichy Jan, 327
Czarnul Paweł, 207
Czerwiński Janusz, 815
Czyżewski Andrzej, 135, 439
Demkowicz Jerzy, 241
Dereniowski Dariusz, 725
Diarra Madian dit Tiéman, 215
Dobrzański Lech, 815
Dorsz Maciej, 569
Durślewicz Jerzy, 645
Dzierżko Jerzy, 815
Fiszer Marcin, 207
Franczak Michał, 899
Frączak Marcin, 207
Fuszara Marcin, 733, 755
Gajewska Małgorzata, 279
Gajewski Sławomir, 279
Gajos Piotr, 447
Galwas Bogdan A., 627
Gawroński Sławomir, 591
Gierłowski Krzysztof, 635
Gierszewski Tomasz, 455
Górski Grzegorz, 51, 463
Grabowska Anna, 645
Grabowska Teresa, 899
Grzybkowski Maciej J., 339
Gwoździńska Agnieszka, 215
Hasse Lech, 1023, 1029
Hermanowicz Ewa, 143
Hetmański Marek, 597
Illeczko Marek, 1023
Jacymirski Michał, 153
Jakieła Jacek, 199
Jakusz Jacek, 793
Jamous Samer, 471
Janczewski Robert, 743
Jendernalik Waldemar, 793
Jurski Janusz, 287
Juszczyszyn Krzysztof, 257, 899
Kaczmarek Andrzej, 135, 439
Kaczmarek Jerzy, 215, 221
Kaczmarek Sylwester, 479, 487
Kadłubowski Wojciech, 799
Kalicka Renata, 169, 183
Katarzyniak Radosław P., 227, 661
Katulski Ryszard J., 347, 355, 363, 371
Kiedrowski Andrzej, 355
Kiss Bálint, 37
Klajbor Tomasz, 295
Klawikowski Michał, 207
Kłosowski Miron, 807
Kmiecik Alina, 671
Kokosiński Zbigniew, 747
Koprowska Monika, 233
Kordalski Wiesław, 815, 825
Kosiński Andrzej, 379, 387
Kosowski Adrian, 495, 733, 755
Kostek Bożena, 975
Kostrzewa Marek, 679
Kotus Józef, 135
Koziel Sławomir, 833, 843
Kozłowski Wojciech E., 577
Kraśniewski Andrzej, 1
Krawczyk Henryk, 19. 59
Krawczyk-Bryłka Beata, 561
Krzysztofek Kazimierz, 603
Krzyżniewski, 439
Kuchta Jarosław, 687
Kuszewski Adrian, 611
Kwarciany Krzysztof, 747
Kwiatkowski Sławomir, 447
Lasota Henryk, 707, 1035
Lebiedź Jacek, 915, 923, 931
Lipiński Piotr, 153
Lubacz Józef, 1
Łapienko Konrad, 611

Malak Henryk. 67
Malek Krzysztof, 303
Malinowski Sławomir, 807
Malafiejski Michał, 755
Marczak Andrzej, 311
Matusz Paweł, 319
Maza Jacek, 653
Maziewski Przemysław, 161, 439
Mazur Leszek, 503
Mikołajski Michał, 363, 395
Molisz Wojciech, 511
Moszyński Marek, 241, 883
Mrugalski Tomasz, 519
Nahotko Marek, 941
Narloch Marcin, 479
Nawrocki Jerzy, 249, 569, 653
Nawrocki Jerzy, 699
Neubauer Wojciech, 73
Niedostatkiewicz Marek, 85
Niski Rafał, 311
Nowak Krzysztof, 487
Nowicki Krzysztof, 543, 635, 1011
Odya P., 439
Owczarek Rafał, 661
Paja Wiesław, 177
Pakieser Jerzy, 327
Paliświat Bartosz, 249
Pankiewicz Bogdan, 851
Pardo Katarzyna, 577
Partyka Andrzej, 241
Pawałowski Piotr, 699
Pawlik Arkadiusz, 135
Pawłowski Wiktor, 403
Pazio Marcin, 949
Peliwo Renata, 653
Pepol Stefan, 765
Piasecki Józef, 1029
Piechówka Maciej, 619
Pietrenko-Dąbrowska Anna, 183
Pilek Bogusław, 387
Piotrowski Maciej, 199, 561
Piwakowski Konrad, 765
Pluciński Jerzy, 93
Polowczyk Michał, 815
Pomianek Bartosz, 199
Pospiech Krzysztof, 569
Przyborski Marek, 413
Rak Jacek, 527
Rak Remigiusz J., 627
Rakiej Jacek, 207
Ramczykowska Katarzyna, 207
Rojewski Mirosław, 125, 143
Ruciński Andrzej, 101
Rumiński Jacek, 191
Rutkowski Dominik, 303
Rykaczewski Roman, 957
Rypułak Andrzej, 135
Sachse Paweł, 257
Sadowski Jarosław, 371
Sawzdargo Rafał, 233
Seliga Wojciech, 965
Skrygulec Artur, 101
Skwarek Tomasz, 645
Smoleński Lech, 439, 535
Sobociński Adam, 125
Solecki Mariusz, 851
Sosnowski Maciej, 303
Spiralski Ludwik, 1029
Stankiewicz Marcin, 923
Stasieńko Justyna, 585
Stefański Jacek, 265
Stefański Tomasz, 825
Stencel Artur, 503
Stepnowski Andrzej, 883
Stępniak Marta, 931
Studański Ryszard, 417
Suchcicki Krzysztof, 207
Sworowski Rafał, 707
Szczepanski Stanisław, 843
Szcześniak Piotr, 859
Szcześniak Władysław, 859
Szczuko Piotr, 439, 975
Szejko Stanisław, 619
Szpankowski Wojciech, 17
Szwoch Mariusz, 965, 993
Szwoch Wioleta, 983, 993
Światowiak Jacek 543
Świecciak Daniel R., 503
Uhl Tadeus, 1011
Vajta László, 37
Vámos Gábor, 37
Waloszek Wojciech, 1003
Weisbrodt Ryszard, 503
Werla Marcin, 699
Wesołowski Marek, 425
Wielgus Michał, 59
Wierzba Paweł, 93
Witkowski Tomasz, 777
Woźniak Józef, 19, 73, 287, 295, 319
Wroński Marek, 865, 875
Wróbel Michał, 221
Wrzesień Mariusz, 271
Zawadzki Michał, 907
Zieliński Mariusz, 425
Zientalski Marian, 439
Żwan Paweł, 135