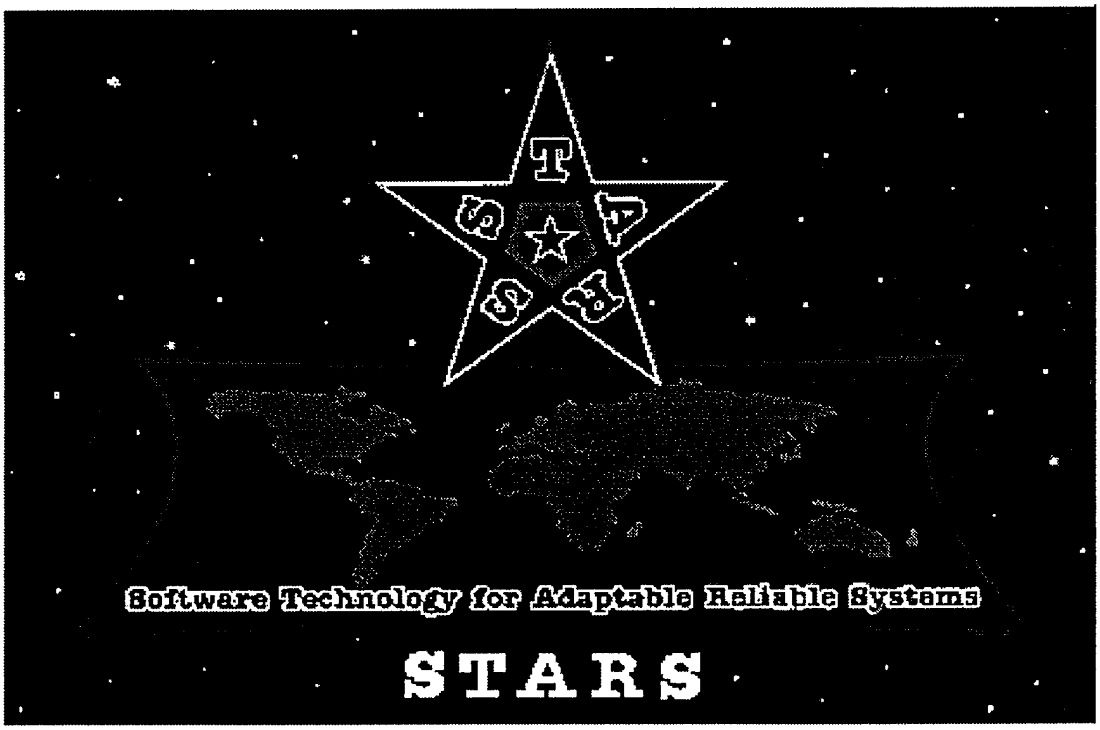
TASK: PA09
CDRL: J001
07 April 1995

Process Presentation using the World Wide Web Lessons Learned



STARS-AC-J001/001/00A

TASK: PA09
CDRL: J001
7 April 1995

INFORMAL TECHNICAL REPORT

For

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

Process Presentation using the World Wide Web
Lessons Learned

STARS-AC-J001/001/00A
7 April 1995

Data Type: Informal Technical Data

CONTRACT NO. F19628-93-C-0130

Prepared for:
Electronic Systems Center
Air Force Systems Command, USAF
Hanscom, AFB, MA 01731-2816

Prepared by:
Unisys Corporation
12010 Sunrise Valley Drive
Reston, VA 22091

REPORT DOCUMENTION 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
07 April 1995

3. REPORT TYPE AND DATES COVERED
Informal Technical

4. TITLE AND SUBTITLE
Process Presentation using the World Wide Web
Lessons Learned

5. FUNDING NUMBERS
F19628-93-C-0130

6. AUTHOR(S)
Ed Guy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Unisys Corporation
12010 Sunrise Valley Drive
Reston, VA 22091-3499

8. PERFORMING ORGANIZATION REPORT NUMBER
CDRL NBR
STARS-AC-J001/001/00A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Air Force
ESC/ENS
Hanscom AFB, MA 01731-2816

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
J001

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution “A”

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Effective transition of complex domain engineering technologies requires the generation of different kinds of documentation to meet the needs of different audiences. Addressing the needs of a potential adopter who is unfamiliar with domain engineering technologies may require creation of descriptive material that introduces the technologies and provides guidance toward their adoption. For practitioners more formal technical descriptions of the methods and tools may be in order. Integrating such process descriptions into a hypertext web makes it practical to create a combined presentation that communicates different process abstractions to readers who need different views of the process.

14. SUBJECT TERMS

15. NUMBER OF PAGES
13

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
SAR

TASK: PA09
CDRL: J001
7 April 1995

Data Reference: STARS-AC-J001/001/00A
INFORMAL TECHNICAL REPORT
Process Presentation using the World Wide Web
Lessons Learned

Table of Contents

Table of Contents

TASK: PA09
CDRL: J001
7 April 1995

Data Reference: STARS-AC-J001/001/00A
INFORMAL TECHNICAL REPORT
Process Presentation using the World Wide Web
Lessons Learned

Abstract

Effective transition of complex domain engineering technologies requires the generation of different kinds of documentation to meet the needs of different audiences. Addressing the needs of a potential adopter who is unfamiliar with domain engineering technologies may require creation of descriptive material that introduces the technologies and provides guidance toward their adoption. For practitioners more formal technical descriptions of the methods and tools may be in order. Integrating such process descriptions into a hypertext web makes it practical to create a combined presentation that communicates different process abstractions to readers who need different views of the process.

1.0 Introduction

The Unisys STARS Team has developed a World Wide Web (WWW) presentation of the Army STARS Demonstration Project's domain engineering process and has prototyped a capability to automate the generation of similar presentations. The goal of the work is to demonstrate how the Web can be used to facilitate understanding and use of the process.

2.0 Objectives

1. **Communicate effectively with potential users** of the process by describing process steps, showing examples of work products, and illustrating the use of SEE tools to support enactment.
2. **Facilitate technology transition** by providing WWW access to a process description that takes advantage of hypermedia presentation technologies.
3. **Provide an example** for the STARS program, the Software Reuse Initiative, and other technology based programs of how effective on-line process presentation can be accomplished.

3.0 Communicating Effectively with Potential Users

The Unisys STARS Team's approach to on-line process presentation is based on a simple assertion,

> "Print documents are not the best means of communicating the use of new, complex methods and tools."

A large body of information about the domain engineering process, including detailed text, descriptive and definitive graphics, definitions of terms, indices, and references to additional materials, is required to communicate with the many potential beneficiaries, adopters, and practitioners of the process. The ability of a particular reader to locate needed information in a comprehensive print document would be hampered by the inherently serial nature of the medium, and by the degree to which the presentation is consistent with the reader's needs. Multiple sets of documentation could be produced, tailoring each set to the needs of a particular reader; the production and maintenance of multiple documentation sets increases the cost of the documentation and the potential for inconsistency.

Hypermedia technologies make it possible to combine different kinds of process information from separate sources in a single, integrated presentation. Informal, text-based descriptions can be used to convey high-level, tutorial information with process beneficiaries and potential adopters. Formal process definition methods can be used to communicate with process engineers responsible for instantiating and executing the process. Structural relationships among the descriptions can provide a basis for using automated techniques to correlate and cross-reference the descriptions.

3.1 Information Engineering

The Army STARS Demonstration Project's domain engineering team is creating a *"Domain Engineering Guidebook"* that combines a variety of formal and informal descriptive techniques to define their process. The techniques provide different representations of the process, and are generated with different tools:

> *Textual descriptions* - introduce topics, provide rationale for decision making, provide examples of actual work products, explain dependencies on other processes, etc. (produced with FrameMaker)
>
> *Process trees* - depict task hierarchies and provide referential context for all users of the definition (produced with FrameMaker)
>
> *IDEF0 diagrams* - model task relationships and communicate concepts among project members (produced with Design/IDEF)
>
> *Process Breakdown Structures (PBS)* - organize detailed task information and support automated process analysis (produced with Process Mapper)

These representations convey a hierarchical definition of the domain engineering process. The *Guidebook* is similarly structured, with sections and sub-sections of the document reflecting the decomposition of the process trees, the IDEF0 model, and the Process Breakdown Structures. This common hierarchical structure suggests a straight-forward mapping of *Guidebook* headings to IDEF0 box labels and PBS step names, however, a great deal of variation exists in the names used to identify process activities in the different representations. *Guidebook* headings tend to be more expressive than IDEF0 box names, which are relatively terse; PBS step names are single strings whose length and format are restricted. While variation could be minimized by establishing restrictive naming conventions, such conventions could also restrict communication with a target audience. A means of establishing a relationship among the different names is needed.

The domain engineering team has also created a *process packet* for each activity in the domain engineering process. The packets include textual descriptions, process trees, IDEF0 diagrams, and Process Mapper reports. Example work products, tool usage scenarios, and copies of pertinent technical papers may also be included, as needed to guide engineers in the performance of process tasks. The packets are provided to engineers in printed form by extracting process data from various tools in the Unisys STARS Software Engineering Environment (SEE).

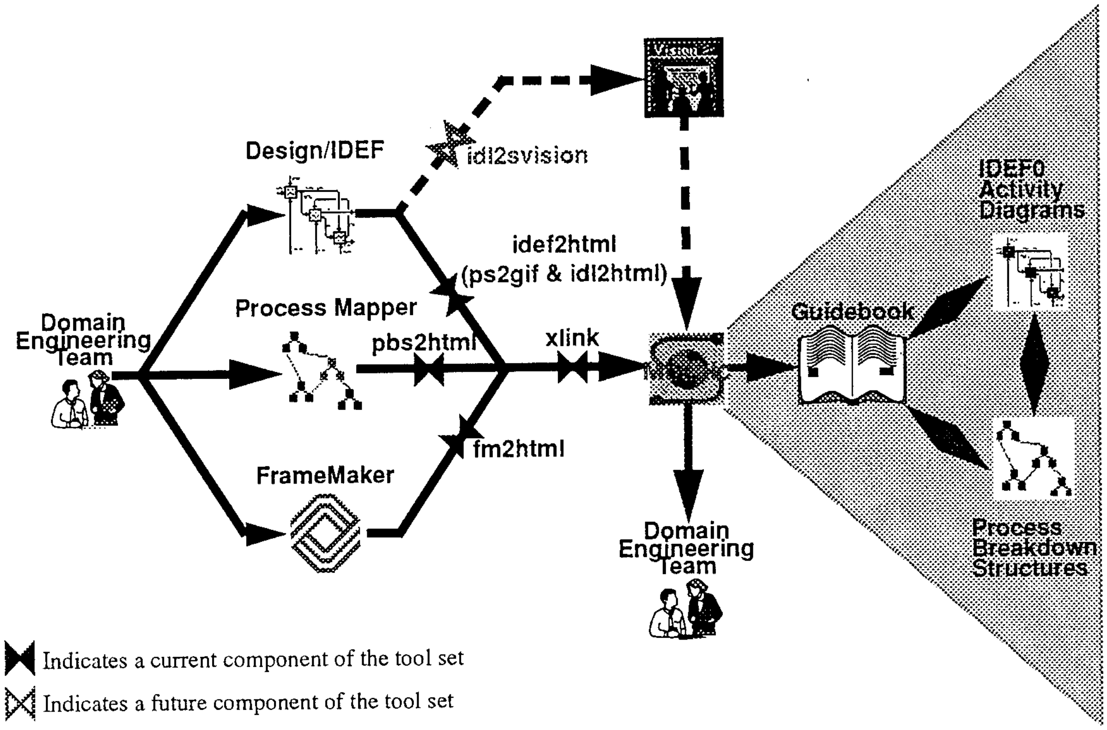
3.2 Web Generation

Since the *Guidebook* and its underlying process definition are still evolving, a one-time translation of the *Guidebook*, IDEF0 models, etc. to HTML would be of limited value. To react quickly to changes a repeatable web generation process is needed. The process is supported by an automated *HTML Process Presentation Tool Set*[1], which takes advantage of the openness of the SEE to generate a WWW process presentation from the same process data used to construct the *Guidebook* and the *process packets.*

This capability has been created by combining tool export utilities, source format translators, and Unisys-developed HTML generators. This tool set is combined to produce a process presentation

[1] Data Reference: STARS-AC-A022/004, ASSET_A_869

that makes each view (text, IDEF0, PBS) accessible from each other view. Figure 1 describes the conversion process. Table 1 identifies the primary components of the tool set.



⧓ Indicates a current component of the tool set

⋈ Indicates a future component of the tool set

Figure 1: Automated Web Generation

Table 1: Tool Set Components	
fm2html	a public domain utility, generates HTML files from FrameMaker source document format
idef2html	a set of Unisys-developed tools, generates HTML files from Design/IDEF output formats
idl2svision	(under development) a Unisys-developed tool, generates HP SynerVision process templates from an IDL (IDEF Description Language) representation of the process.
pbs2html	a Unisys-developed tool, generates HTML files from Process Mapper report format
xlink	a Unisys-developed tool, inserts cross-referenced HTML anchors into the HTML files, based on structural information contained in the individual models

The HTML presentation that results from this process allows a newly assigned domain engineer to browse the on-line Guidebook's table of contents to find the section describing an assigned task. After clicking on the section heading the engineer is presented with text and graphics that describe the intent and general flow of the activity. By clicking on an icon embedded in the Guidebook text, the engineer can browse the IDEF0 diagram for the same task, using either activity boxes within the diagram or an attached tree structure to navigate the IDEF0 model. From either the IDEF0 diagram or the Guidebook the engineer can access the formal Process Breakdown of the task to review its purpose, a description of the work products used and produced, its entry and exit criteria, and more. When integrated with SynerVision, the presentation can be accessed by requesting help from within the task structure, with appropriate context already established.

3.3 Evolution of the Presentation and Tool Set

3.3.1 Initial Experiments

The initial product of the effort consisted primarily of HTML representations of the physical process definition artifacts, forming three separate hypertext webs: one for the *Guidebook*, one for the IDEF0 model, and one for the PBS model. A simple translation of the *Guidebook* from FrameMaker source format to HTML was performed, using the public-domain *fm2html* translator. PostScript output from Design/IDEF was used to build an image-mapped HTML page for each activity. A process tree, showing hierarchical context, was derived from the PostScript and displayed along with the IDEF0 diagram. Navigation of the process hierarchy was accomplished by clicking on IDEF0 activity boxes or tree branches. HTML versions of Process Mapper reports were created and linked to the IDEF0 diagrams.

3.3.1.1 Presentation Style

The purpose of the initial product was to demonstrate how hypermedia technologies could be used to effectively guide engineers through the performance of domain engineering activities. With that technical audience in mind, an *annotated process model* which presented a formal process definition (IDEF0 diagrams and Process Breakdown Structures) in graphical form, augmented with textual information, seemed an appropriate form of presentation. Such an approach would ensure consistency of the documentation with the formal process definition. The *Guidebook*, on the other hand, reflected a different sort of organization in which textual descriptions were augmented with selected graphics. This *illustrated process guidebook*, though lacking the consistency and completeness of a formal definition, presented a well-organized overview of the process, aimed at managers and others who need to understand domain engineering concepts and a general approach for adopting the process.

> An informal process description, like that found in the *Guidebook*, can communicate with various audiences, but can rarely be used reliably as the basis for automated analysis of the information it conveys. In contrast, a model-based presentation, while useful for automated analysis and potentially appropriate as a technical reference for an experienced engineer, may be ill-suited for communicating with non-technical audiences. Initial feedback made it clear that any attempt to automate the generation of an HTML process presentation would have to support incorporation of both, formal definitions, and original creative material.

3.3.1.2 Guidebook Conversion

Automatic translation of FrameMaker documents to HTML proved somewhat problematic. Figures that were not enclosed in anchored frames were ignored by the translator. Figures that were enclosed in nested anchored frames were ignored by the translator. Tables that were enclosed in anchored frames were ignored by the translator. Information regarding these problems was passed on to the *Guidebook* authors.

> The capabilities of the *fm2html* translator are limited. Newer versions of *fm2html* are available, as are other utilities for converting FrameMaker documents to HTML[2]. We plan to experiment with several of these utilities and recommend the same to others.

Most small or dense figures translated poorly to screen presentation. Typical 72-dot-per-inch screen resolutions do not adequately represent images that were constructed with 300-dot-per-inch resolution printers in mind. We were able to compensate somewhat for the loss of resolution by applying a common magnification factor as the figures were converted from PostScript to GIF format (using the *-xdpi* and *-ydpi* arguments to the *fm2html* translator). The efficacy of this adjustment was limited by the resulting size of larger graphics.

> It is reasonable to expect that no automatic translator will convert every FrameMaker element accurately. Organizations should develop process documentation guidelines that account for both, print and CRT presentation, and for the automatic generation of HTML from document source format.

3.3.1.3 Model Conversion

Generation of HTML from the formal process definitions was highly dependent on the output formats of the modeling tools. Interpretation of Design/IDEF's PostScript output required significant PostScript programming expertise and established an unacceptable dependency on that particular tool. When a Unix version of Design/IDEF, providing IDL[3] export capability, became available our dependency on the tool was minimized.

> The Unisys STARS SEE has been constructed to the greatest extent possible using tools that adopt open standards. The existence of data exchange interface standards, such as IDL, significantly enhances the ability of an automated environment to adapt to technological changes. The adoption of those standards by a tool vendor strengthens the vendor's position in the tool selection process. We are working with various process engineering support tool vendors to encourage migration to standards-based tool interaction and welcome the interest of additional parties.

[2] **http://www.w3.org/hypertext/WWW/Tools/Word_proc_filters.html#Framemaker**

[3] IDEF Description Language

3.3.2 Integrating the Approaches

Mechanisms were developed for integrating the formal process descriptions with the creatively produced *Guidebook*. The HTML generators were reorganized and a HTML scanner was added to cross-reference the three webs (Figure 1).

3.3.2.1 fm2html

As previously stated, some problems were encountered when using this coverter to generate HTML from FrameMaker source format (section 3.3.1.2). Typically, some manual post-conversion manipulation of the output was required to optimize the on-screen appearance of the document. In spite of these inconveniences, the utility proved suitable to our experimental needs; no attempt has been made to modify or enhance the *fm2html* converter (refer to section 3.3.1.2).

3.3.2.2 idef2html

This script set uses X Window System utilities (*ps2gif*) to translate Design/IDEF's PostScript output to a separate GIF-formatted image of each IDEF0 diagram. Using Design/IDEF's IDL export format, the script *idl2html* extracts information about the organization and placement of diagram elements, and uses that information to generate image mapped IDEF0 diagrams and process trees.

The readability of the IDEF0 diagram images (and, in fact, all graphics) produced by the *ps2gif* conversion is highly dependent on the text fonts chosen when the diagrams were originally produced. The selection of large, bold, san-serif fonts (Helvetica, Arial, Avant-Garde) makes the resulting images easy to read, but can limit the amount of information that can be included in the diagram. Imposition of a font standard after the diagrams are produced can necessitate changing the layout of the diagrams to accommodate larger text. Logic has been incorporated into the *idl2html* script set which attempts to improve the images by changing font styles and sizes during the conversion process. The relative improvement depends on the resolution of the viewing screen and the layout of the original diagram. Logic was also added to shade the image-mapped portions of the diagram, making it easier to determine which IDEF0 boxes are decomposed, which arrows have formal definitions, etc.

3.3.2.3 pbs2html

Process Mapper can generate separate reports that describe Process Breakdown Structures: process steps (activities), work products, roles, and conditions. The *pbs2html* script set uses these reports (stored in ASCII format) to create a separate HTML file for each step, work product, etc., and establishes internal cross-references among them.

The creation of multiple HTML files reduces the amount of time required to load pertinent information when viewing the presentation.

3.3.2.4 xlink

This special-purpose HTML scanner searches the HTML *Guidebook* for section headings (<Hn> tags) and attempts to locate an HTML IDEF0 diagram whose title matches, or differs only in capitalization or the substitution of *underscore* characters for *space* characters, with the section heading. If a match is found, an HTML reference to the IDEF0 diagram is inserted into the *Guidebook*; if not, *xlink* will attempt to locate the HTML IDEF0 diagram whose node number is specified in a synonym table (section 3.3.2.5) and will insert an appropriate reference.

xlink also attempts to locate an HTML Activity Definition (from Process Mapper) whose name matches, or differs as described above, with the guidebook section heading. If a match is found an HTML reference to the Activity Definition is inserted into the Guidebook; if not, *xlink* will use the synonym table to locate an appropriate reference.

3.3.2.5 Synonym Table

To circumvent problems associated with attempting to maintain complete consistency in the naming conventions used in each process description, a synonym table is used to correlate equivalent hierarchical elements. Each text line in the manually created *SynonymTable* file correlates between a Guidebook section heading (field 1), an IDEF0 diagram node number (field 2), and a PBS Activity Description (field 3)[4]. A complete entry must be created when either the title of an IDEF0 diagram or the name of a PBS Activity Description does not match the corresponding Guidebook section heading. Consistent use of canonically formed names in the Guidebook, the IDEF0 model, and the PBS model can eliminate the need for the table.

Partial SynonymTable

```
Plan for domain engineering A1 Plan_for_domain_engineering
Define domain engineering objectives A11 Define_domain_eng_objectives
Identify candidate domains A12 Identify_candidate_domains
Develop a domain information sources list A121 Dev_domain_info_sources_list
Develop a domain lexicon A122 Develop_domain_lexicon
Synthesize candidate domains A123 Synthesize_candidate_domains
```

The table could be expanded to establish correlation among other process definition elements, for example SynerVision task hierarchies and AutoPlan tasks.

[4] Syntax for the SynonymTable file is described in
Data Reference: STARS-AC-A022/004/00
VERSION DESCRIPTION DOCUMENT
HTML Process Presentation Tool Set
Version 0.5
Prototype

3.3.3 Application

The final phase of activity focused on generating a complete WWW presentation of the "Planning for Domain Engineering" sub-process. Feedback from early experimentation and integration activities was provided to the *Guidebook* authors, who reviewed the completeness and consistency of the textual descriptions, IDEF0 models, and Process Breakdown Structures. The tool set was used to generate an HTML web presentation of the "Planning for Domain Engineering" process description. Since the current version of the tool set only creates references for text that is part of an HTML heading, references to Glossary entries and Appendices were added manually.

Feedback from the Demonstration Project has been extremely positive and indicates that WWW presentation of the domain engineering process may eliminate the need for *process packets* in printed form. The *Guidebook* authors are evaluating the impact of WWW presentation on the organization and construction of the document, which will continue to exist in print and WWW forms.

4.0 Illustrating Enactment Support

A **process-driven** approach to supporting process performance distinguishes itself from tool-centered approaches in the way tools are used. In a tool-centered environment a set of tools is made available to the user to support process activities. The tool set may be limited or invocation restricted based on the user's current role or process context. Typically, the user must take specific action to invoke a tool, which requires a knowledge of the tool's location, calling sequence, etc. In a process-driven approach appropriate tools are automatically invoked based on the task being performed. Tool locations are transparent to the user; invocation sequences are determined from task context.

HP SynerVision, which fills the role of Task Manager in the Unisys STARS SEE, has been used to provide automated support for a portion of the domain engineering process. SynerVision represents the process as a hierarchy of tasks, presented to the user as an indented outline. SynerVision *templates* have been created to invoke appropriate domain engineering tools when the engineer signals commencement of a task. The *templates* also include menus of *actions* that allow the engineer to view and manipulate work products within the bounds of the task being performed. The Unisys Reuse Library Framework (RLF) is invoked, via SoftBench messages, to assist the engineer with domain modeling activities. SoftBench messages are exchanged with a Product Manager tool class to control repository check-in/check-out processes. These exchanges are transparent to the engineer, except when the invoked tool requires information that cannot be derived from the task context. On-line guidance is provided via a "HELP" action, which invokes Mosaic to present the on-line *Domain Engineering Guidebook* in an appropriate hierarchical context. Access to technical papers, tutorials, tool usage scenarios, personal notes, and other on-line documentation could be provided through the same mechanism.

The process information used to develop the SynerVision implementation was extracted manually from the printed *Guidebook*; some difficulty was encountered responding to changes in the document. Our experience in the automatic generation of regularly structured process presentations from consistent process definitions suggests that similar techniques could be used to generate SynerVision process templates. The consistency gained through such automation would also make it easier to establish links between the SynerVision implementation and the WWW presentation.

5.0 General Observations

5.1 Benefits of Automation

5.1.1 Cost Savings

Converting the Guidebook from FrameMaker to HTML, generating PostScript and IDL outputs for the IDEF0 model, and generating the necessary Process Mapper reports takes about 30 minutes. Running the tool set on the selected sub-process takes about 25-30 minutes. The tool set automatically creates HTML for the IDEF0 and Process Mapper models, process trees and image maps for the IDEF0 diagrams, and all the HTML references needed to integrate the presentation. The tools run in the background, requiring no user interaction. The tool set's 1/2-hour execution time compares favorably against the following short list of activities that would otherwise have to be performed manually:

> Creating image maps: 10 minutes per image segment x 144 segments in sample web = 24 hours
>
> Cross-referencing process representations: 25-30 hours
>
> Generating process packets: 1/2 hour per packet x 340 packets in Application Engineering Process = 170 hours
>
> We recommend use of these tools to organizations developing process definitions using IDEF0 and ProcessMapper.

5.1.2 Taking Advantage Standards

5.1.2.1 Image Conversion

Creating mapped images manually is a multiple-step process involving translation of images from one format to another, determining and recording image coordinates, and testing the results. The tool set uses public-domain image conversion programs included as part of the X Window System (the *bin/pbm* subdirectory) to convert PostScript images to GIF format.

> The use of these utilities virtually eliminates the potential for manual introduction of error in the conversion process. Any organization using the X Window System to produce graphics for on-screen presentation should become familiar with the *pbm* utilities.

5.1.2.2 Analyzing the Process Definition

The IDEF Description Language (IDL) provides a consistent description of IDEF0 diagram information that can be used to drive definition of other process engineering activities. SynerVision process templates, AutoPlan projects, and Process Mapper process breakdown structures are among the hierarchically organized process representations that show potential for being generated automatically, at least partially, from IDL. The existence and utility of this standard are important factors in the incremental progression of process-driven environments from tool-dependent to standards-supported to standards-based approaches.

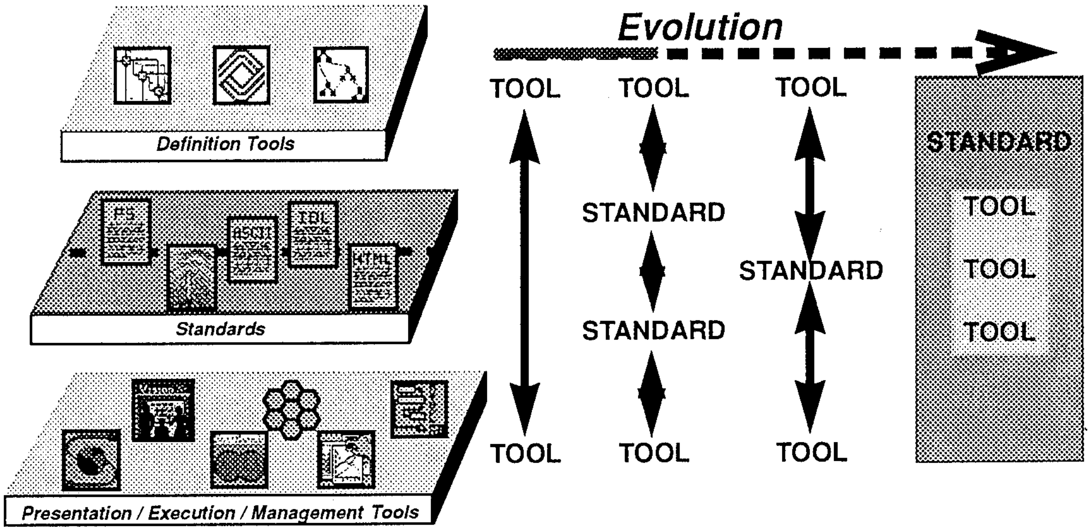


Figure 2: Evolution of Process Tool Integration

Using one process representation to aid in the development of another can guarantee consistency between the two representations. We are aware of no single process definition method that would support universal transformation to meet all process engineering needs. We have learned that simply attempting to cross-reference one representation to another exposes discrepancies in the completeness and the consistency of the process definition that can easily be corrected.

5.1.3 Information Engineering

Effective web communication involves more than just writing good documents or writing good HTML. Preparing documentation for on-line viewing involves a different set of considerations than those involved in preparing documentation for print. The presentation spaces are of different sizes and support different image resolutions, both of which can dramatically impact the effectiveness of the presentation. Hypermedia technology makes it possible to combine text, graphics, audio, and video to produce exciting presentations. Not every reader will be able to take advantage of all media. Some HTML browsers don't support the use of *alternate text* for graphics, so any information represented only in graphical form can be lost. The same is true of audio/video-based information.

Organizations considering HTML presentation of complex information should carefully consider the needs of the target audience when developing the presentation. Traditional print-based approaches should be augmented by the additional capabilities of on-line presentation technologies (hypertext association of separate documents, availability of audio and video information, etc.). The approach should also be tempered by the special problems of the technology (image resolution, network interconnection reliability, etc.)

5.1.4 Shell Programming

This experience has confirmed that shell programming, whether C shell, Bourne shell, Korn shell, or other is not a very effective means of creating a durable software product. This, along with our work in automated process enactment and tool integration, reveals that idiosyncracies among shell implementations in a heterogeneous development environment encourage the use of quick "hacks" to circumvent problems. The result is a code set that, although adequate as a prototype, does not readily lend itself to adaptation.

> A standard, interpretive command/scripting language for Unix systems is needed. The language should incorporate the capabilities of the numerous existing file, string, and numerical manipulation utilities, in a consistent syntactic and semantic framework.

6.0 Development Potential

6.1 The Tool Set

At present, the *HTML Process Presentation Tool Set* can only be described as a prototype. It behaves predictably when used to convert a coherent set of *Guidebook*/IDEF0/PBS models, as is the case for the *Guidebook* authors. It will not convert stand-alone IDEF0 models to HTML, nor will it serve as a general-purpose HTML link generator to automatically generate references to glossaries or bibliographies. A partial list of potential modifications follows:

1. Improve FrameMaker-to-HTML conversion by upgrading from *fm2html* to *WebMaker*
2. Separate the tool set into individual conversion utilities
    a. an independent IDL-to-HTML generator
    b. an independent IDEF0 diagram shading utility
    c. a general-purpose utility for creating HTML references
    d. an independent PBS-to-HTML generator
3. Add a user interface for specifying configuration options, layout preferences, etc.
4. Minimize dependency on the SynonymTable
5. Reformat the Process Mapper reports
6. Create a name for each anchor that corresponds to the text associated with the anchor

6.2 The Technology

The success of this effort suggests additional applications of the technology:

1. Extensions

a. Develop an IDL-to-SynerVision generator

b. Develop an IDL-to-AutoPlan generator

c. Develop a utility to create HTML image maps for RLF models

d. Integrate tool usage scenarios with the process description

2. Development of on-line methods tutorials. For example,

a. IDEF0

b. Process Breakdown Structures

c. Organization Domain Modeling

d. GenVoca

3. Development of additional process presentations

a. Process engineering processes

1. Process definition

2. Process instrumentation

3. Process implementation

d. SEE integration processes

e. Application engineering processes

TASK: PA09
CDRL: J001
7 April 1995

Data Reference: STARS-AC-J001/001/00A
INFORMAL TECHNICAL REPORT
Process Presentation using the World Wide Web
Lessons Learned

Distribution Statement "A"
per DoD Directive 5230.24
Authorized for public release; Distribution is unlimited.

This document, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "A" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24) unless otherwise indicated. Sponsored by the U.S. Advanced Research Projects Agency (ARPA) under contract F19628-93-C-0130, the STARS program is supported by the military services, SEI, and MITRE, with he U.S. Air Force as the executive contracting agent. The information identified herein is subject to change. For further information, contact the authors at the following mailer address:
delivery@stars.reston.paramax.com.

The contents of this document constitute technical information developed for internal Government use. The Government does not guarantee the accuracy of the contents and does not sponsor the release to third parties whether engaged in performance of a Government contract or subcontract or otherwise. The Government further disallows any liability for damages incurred as the result of the dissemination of this information.

In addition, the Government (prime contractor or its subcontractor) disclaims all warranties with regard to this document, including all implied warranties of merchantability and fitness, and in no event shall the Government (prim contractor or its subcontractor) be liable for any special, indirect or consequential damages or any damages whatsoever resulting from the loss of use, data, or profits, whether in action of contract, negligence or other tortious action, arising in connection with the use of this document.

TASK: PA09
CDRL: J001
7 April 1995

Data Reference: STARS-AC-J001/001/00A
INFORMAL TECHNICAL REPORT
Process Presentation using the World Wide Web
Lessons Learned

Principal Authors():

Ed Guy Date

Approvals:

Program Manager *Teri F. Payton* Date

(Signatures on File)