

AD-A240 055



2

Natural Language for Problem Solving Systems Final Report

Principal Investigator: Kathleen R. McKeown

PI Institution: Columbia University

PI Phone Number: 212-854-8194, 212-854-2736 (messages)

PI Electronic Mail Address: MCKEOWN@CS.COLUMBIA.EDU

Contract Title: Natural Language for Problem Solving Systems

ONR Contract Number: N00014-82-K-0256

Period Covered: 4/1/87-3/31/89



This document has been approved
for public release and sale; its
distribution is unlimited.

58 91-08015



Table of Contents

1	Productivity Measures	2
2	Summary of Technical Progress	3
3	Detailed Summary of Technical Results	4
	3.1 Natural Language Interpretation for expert systems	4
	3.2 Automated Tutoring for Extending User Expertise	5
	3.2.1 Domain Knowledge Source	6
	3.2.2 Explanatory Knowledge	7
	3.3 Expert System Explanation	8
4	Publications, Presentations, and Reports	8
5	Honors and Awards	11
6	Research Transitions and DoD Interactions	12
7	Software and Hardware Prototypes	13



Accession For		J
NTIS	ORAS	
DDIC	TAB	
Unat	Unat	
Justification		
By		
Distribution		
Availability Codes		
Dist	Special	
A-1		

Statement A per telecon
Dr. Andre VanTilborg ONR/
Code 1133 Arlington, VA 22217-
5000
NWWW 9/4/91

1 Productivity Measures

Refereed papers submitted but not yet published: 0

Refereed papers published: 11

7 (conference proceedings)

3 (journal)

2 (workshop)

Unrefereed reports and articles: 9

Books or parts thereof submitted but not yet published: 0

Books or parts thereof published: 2

Patents filed but not yet granted: 0

Patents granted: 0

Honors received: 8

Prizes or awards received: 1

Promotions obtained: 2 (promoted to Associate, granted tenure)

Graduate students supported: 3

Post-docs supported: 0

Minorities supported: 3 women

2 Summary of Technical Progress

Over the course of the contract, we have had results in three separate projects: natural language interpretation for expert systems (covered under the previous contract), goal oriented explanation generation for expert systems, and generation for intelligent tutoring using domain and explanation knowledge sources (covered under the previous contract and the renewal grant). We have completed implementation of semantics for a natural language interface to expert systems. This has included development of a meta-structure for the underlying rule base derived from linguistic classification of verbs, an algorithm for semantic interpretation using the meta-structure representation, and a facility for completing semantic parses over dialog. We have also run a transportability test, moving the system from the domain of income tax advising to the domain of interview scheduling. In our intelligent tutoring system, we have developed a semantic taxonomy for relations between plans, implemented and tested our plan analyzer, and begun design on flexible interaction between schemas (structures in our explanation knowledge source that control generation of content). Finally, we have completed several evaluation studies of our fully implemented system for generating goal oriented explanations.

3 Detailed Summary of Technical Results

Over the course of the contract, we have had results in three separate projects: natural language interpretation for expert systems (covered under the previous contract), generation for intelligent tutoring using domain and explanation knowledge sources (covered under the previous contract and the renewal grant), and the generation of goal oriented explanations for expert systems. In this section, we describe progress in these three areas separately.

3.1 Natural Language Interpretation for expert systems

During its problem solving activity an expert system must gather information necessary for arriving at a problem solution. Expert systems that gather this information interactively from their users typically use a menu to do so. We are involved in research to replace this menu with natural language input. One of the main problems in such an enterprise is the development of a semantic representation that can be used for the expert systems environment and a corresponding parsing algorithm. Due to the unstructured nature of the typical expert system rule base, little information is available about the meaning of predicates of individual rules and their interrelationships and yet, this is exactly the information that is needed to map natural language utterances into corresponding rules and goals in the underlying system.

In previous years, we designed and implemented a representation that is being used to provide a structure for the underlying expert system, based on hierarchical verb categorization. This structure consists of a forest of 13 different hierarchical verb categories and is used to provide a structure for the rules in the tax advising system we have developed. It is also used as the basis for our parsing algorithms. goals in the underlying system.

In previous years, we had designed a preliminary representation that could be used to provide a structure for the underlying expert systems. This structure is based on hierarchical verb categorization. Over the past year, we refined and extended this representation, developing the hierarchies for 13 different verb categories and implementing them as part of the tax advising system we have developed. We have shown how this representation can be used to provide a structure for the propositions in an unstructured system, such as an expert system, and how a parsing algorithm for natural language interpretation can be encoded directly in the hierarchies (Moerdler 88). We have fully implemented the hierarchies and parsing algorithm for statements and yes-no questions.

A second problem in this environment results from the interactive nature of the system. Over the course of a session, the user will pose a goal for the system to solve (usually in the form of a question) and the system, in turn, must ask questions to gather additional information needed for problem solving. From user responses, the system derives facts and places them in working memory to be used for problem solving. Difficulties arise because a single user utterance may not always correspond to a single expert system proposition. Several system propositions may be provided through a single utterance (e.g., in answering a question, the user may provide more information than was explicitly requested thus obviating the need for a future system question). Similarly, the information needed to derive a single system proposition

may be provided over several user utterances, and these need not occur consecutively in the dialog.

The semantic parsing algorithm that we developed is compositional in nature, thus allowing us to derive different propositions from different parts of an utterance. We developed and implemented an algorithm for semantic incompleteness (Moerdler and McKeown 88), that stacks partially derived propositions and completes the derivation when the necessary information arrives later in the session.

Finally, we have extended our parsing algorithm so that it can handle wh-questions in addition to statements and yes-no questions. This involved creating a meta-hierarchy over the domain hierarchies, which describes how it can be accessed. Thus, our system can answer certain wh-questions using the hierarchies alone (i.e., without invoking the underlying expert system). This includes questions which ask about the difference between situations (e.g., "Why can my sister claim her son as a dependent while I can't?") or which ask for definitions (e.g., "Who is considered a legal relative?"). Other questions must be answered by invoking the underlying expert system with a partially instantiated goal. Values which allow the goal to succeed are returned in response.

We have also completed a transportability test. In developing the hierarchies, we have carefully separated domain dependent from domain independent information. The transportability test was designed to show how much of the system must be redone for a new domain. We moved to a radically different domain from our income tax domain, a system that can plan interview schedules. It must receive as input, information about when a potential interviewer would like to meet with the interviewee and for what purposes. The underlying expert system was a planner developed by a student at Columbia. The transportability test was quite successful. We were able to construct the interface for the new domain using entirely the set of hierarchies we had already constructed. These hierarchies were simply extended in depth and in new branches to cover the new domain. The parsing algorithm worked with no changes as expected.

A PhD thesis on this work was completed in March 1990.

3.2 Automated Tutoring for Extending User Expertise

Interactive computing environments are designed to provide supportive resources for a range of users with different expertise and computational goals. Whether simple or complex, all such environments contain an underlying set of functions or constructs with which users accomplish tasks. A problem arises in providing resources through which users can initially learn about the environment and then later extend their expertise. The problem we are studying is *how to provide automated tutoring (or, equivalently, consulting) that extends users' expertise in interactive computing environments.*

We take a user's task centered approach to tutoring in which help given is a direct function of

the current context, users' computational goals, and their knowledge about plans to accomplish such goals in the environment. *How* help is given is a function of knowledge about explanatory strategies that can be used to describe domain knowledge. Thus, our work explores the use of two knowledge sources for language generation: domain knowledge, which in this domain is primarily knowledge about plans and goals, and explanatory knowledge, or strategies for presenting information about plans to the user. This work is being implemented as part of GENIE (GENerated Informative Explanations), an answer generating system that specifically tutors to the current needs of the user in the domain of Berkely UnixTM Mail.

3.2.1 Domain Knowledge Source

In order to be able to explain how to carry out a task (e.g., send mail), our system needs a representation of goals (e.g., send mail) and plans for carrying out those goals (e.g., get into send mode, enter an address, type text of message, type C-D, etc.). In order to tailor that explanation to the user's situation, our system needs a representation of the plans the user already knows about and a representation of the user's current situation (e.g., s/he may already be in send mode). Thus, one major component of GENIE is the representation of system plans and goals (the Expert Model) and user plans and goals (the User Model). In addition to the domain representation, GENIE contains a Plan Analyst which can choose the most appropriate plan for a computational goal in a given context, can determine whether a plan will satisfy a goal, and can compare two plans and identify whether there are any mismatches.

Unlike other work on plans and goals, our representation includes alternative plans that can be used to carry out the same goal and explicit semantic links that describe the differences between these plans. Our work has focused on the development of a semantic taxonomy for these links so that we can systematically represent the different relationships that can occur between plans. This taxonomy has been developed and is currently being used by the Plan Analyst. These links are used to determine when one plan is more appropriate than another. There are two basic types of links that can occur: universal links and domain specific links. Universal links would be relevant in any domain and include temporality (for example, whether a plan is to be carried out now or later) and cardinality (for example, whether a plan is to be applied to a set or a single entity). This information is crucial in determining which plan best satisfies a user's goal. For example, if a user wishes to send mail now and is in read mode, an appropriate plan might involve replying to a previous message rather than undoing actions that put him/her in read mode and getting into send mode.

Domain specific links specify differences in the data types of objects that a plan operates on. For example, the mode in which a command can execute is crucial in determining whether a plan is applicable in the current situation. Whether a user is in unix (and, more specifically, in kshell or cshell) or in mail (and, more specifically, in read or send) mode determines what commands are accessible. Note that these features can be represented hierarchically in terms of specificity and the selection of plans can be made at the most general level possible in this hierarchy. Other examples of domain specific links include restrictions on the object of an action or the recipient of a message. Both universal and domain specific links have been

represented and used by the Plan Analyst in GENIE.

In addition to developing the taxonomy of semantic links, we have also completed implementation of the Plan Analyst and tested it on a larger knowledge base, now covering a good part of the mail domain. We are continuing to expand the size of the domain and are testing GENIE's knowledge representation on a totally different domain: MARVEL, a program development environment developed by Gail Kaiser and her students. Our purpose in these experiments is to test robustness of the representation and to do some complexity analysis of the algorithm.

3.2.2 Explanatory Knowledge

Once the Plan Analyst has selected a plan or plans to describe to the user, GENIE uses schemas to determine what information about the plan to include in a response and how to organize that information. While we have used schemas¹ in the past, our focus in this project is on developing a more flexible framework for their use. In past work, schemas resembled text grammars, which rather rigidly determined what information could come next in a text and were represented using an ATN grammar.

We have developed a new representation for schemas using plans. This allows us to represent preconditions and effects of their use, which can in turn control the invocation of new schemas. We have developed four schemas, **Introduce**, **Remind**, **Clarify**, and **Elucidate**, for GENIE and showed that they could be used in direct response to a question as well as in enrichment for opportunistic teaching of new plans. At this point, we have used our new representation in design for each of these schemas and have implemented and tested our schema selector to experiment with resulting flexibility. Testing with the schema selector shows that every combination of two schemas is possible in a single answer, with the exception of one pair: when **reminding** about a plan, **clarify** or **elucidate** will never be used in addition. This allows for much greater flexibility in producing a variety of answers than we were able to achieve in earlier systems. We are currently working on the implementation of each schema using the new representation so that full content can be produced.

We have also designed a set of filtering rules that is used to weed through the output of the schema and merge or delete propositions when possible. These rules make further tests on the user model and result in tighter, more condensed content. They are designed as tests on schema output rather than as tests on schema steps because they often compare output of more than one step in the schema. An example of a filtering rule would be to exclude a plan substep which the user already knows how to carry out (i.e., this plan substep is represented as a full plan in the User Model).

This work will continue throughout the 1989/90 year under a new ONR grant. Our plans are to complete implementation and testing of the schemas, to implement the filtering rules, and to interface schema output to a Functional Unification Formalism for producing the actual sen-

¹discourse strategies that indicate what information should be included at each point in the text

tences.

3.3 Expert System Explanation

One of our ongoing projects has been the development of mechanisms to tailor an expert system explanation to a user's domain goals. In early years of the contract, we developed mechanisms to derive a user's goal from a discourse segment, to represent different points of view in an underlying knowledge base, and to link user goal and point of view in determining explanation content. This model has been fully implemented in the ADVISOR system, which can provide information about courses and advice about what courses a student can and should take in the upcoming semester. In the last year of the contract, we ran two evaluations of the system. One was as part of the final exam of my natural language course. Students were required to use the system, saving journals of their interaction, and then to answer questions for the final exam about their session with ADVISOR. While some of these questions required the students to think about how the system works, other questions required them to evaluate the system. Journals of the sessions were handed in with the final exam questions. Approximately 80 journals, some of which were quite long, were turned in. A second evaluation was done during spring registration. A terminal with ADVISOR running was provided in the Computer Science registration room. A human consultant was there to help people phrase their questions properly if necessary. Any student registering for courses could use the system. About 40 different students used the system and journals of these sessions were saved.

Of the two evaluations, the one involving the natural language class was more helpful because students answered specific questions about how satisfied they were with system answers and made specific suggestions about changes they would like to see. Because this was part of a final exam, they were motivated (for the most part) to provide thorough answers. In both evaluations, we identified problems with the interpretation portion of the program, noting improvements needed both for syntax and semantics. We found definite ways in which generated explanations could be improved as well: negative answers were not well explained and certain specific questions needed improved responses.

4 Publications, Presentations, and Reports

Publications

- [1] Elhadad, M., Seligmann, D., Feiner, S., and McKeown, K.
A Common Intention Description Language for Interactive Multi-media Systems.
In *A New Generation of Intelligent Interfaces: Proceedings of IJCAI89 Workshop on Intelligent Interfaces*, pages 46-52. Detroit, Mi., August 22, 1989.
- [2] Datskovsky Moerdler, G., McKeown, K. R., and R. Ensor.
Building Natural Language Interfaces for Rule-based Expert Systems.
In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*.
Milan, Italy, August, 1987.
- [3] Datskovsky Moerdler, G. and K. R. McKeown.
Beyond Semantic Ambiguity.
In *Proceedings of AAAI-88*. Saint Paul, Minn., August, 1988.

- [4] Datskovsky Moerdler, G.
Structure from Anarchy: Meta Level Representation of Expert System Propositions for Natural Language Interfaces.
In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 121-8. Austin, Tex., February, 1988.
- [5] Maarek, Y., Smadja, F.
Full Text Indexing Based on Lexical Relations, An Application: Software Libraries.
In *Proc. ACM SIGIR*. Cambridge, June, 1989.
- [6] McKeown, K. R.
Generating Goal Oriented Explanations.
International Journal of Expert Systems 1, Issue 4, :377-95, December, 1988.
- [7] McKeown, K. R. and C. L. Paris.
Functional Unification Grammar Revisited.
In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stanford, Ca., July, 1987.
- [8] McKeown, K.R. and W. R. Swartout.
Language Generation and Explanation.
In J. Traub, B. Grosz, B. Lampson, N. Nilsson (editors), *Annual Review of Computer Science*. Annual Reviews Inc., Palo Alto, Ca., 1987.
- [9] Paris, C.L. and McKeown, K.R.
Discourse Strategies for Generating Complex Physical Objects.
In G. Kempen (editors), *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology and Linguistics*. Kluwer Academic Publishers, Boston/Dordrecht, 1987.
- [10] Smadja, F.
Macrocoding the Lexicon with Co-occurrence Knowledge.
In *Proceedings of the 1st International Workshop on Lexical acquisition*. IJCAI 89, Detroit, August, 1989.
- [11] Smadja, F.
Lexical Co-occurrence, The Missing Link.
Journal for Literary and Linguistic computing 4(3), December, 1989.
Oxford University Press.
- [12] Wolz, U.
Tutoring that responds to users' questions and provides enrichment.
In *Proceedings of 4th International Conference on Artificial Intelligence and Education*. University of Amsterdam, Amsterdam, The Netherlands, 1989.
- [13] Wolz, U.
Intelligent interfaces should be task specific and extend user's expertise.
In *A New Generation of Intelligent Interfaces, IJCAI-89 Workshop*. Detroit, Michigan, 1989.
- [14] Wolz, U., K.R. McKeown and G. E. Kaiser.
Automated tutoring in interactive environments: A task centered approach.
Machine-Mediated Learning 3(1):53-79, 1990.

Technical Reports

- [1] Datskovsky G. and Ensor, J.R..
Director -- An Interpreter for Rule Based Expert Systems.
Technical Report, Columbia University, 1986.
- [2] Datskovsky G., Phelan, G.
Let Me Help You: A Strategy for Answering Natural Language Questions in the Expert Systems Environment.
Technical Report, Columbia University, 1988.
- [3] Elhadad, M. and McKeown, K.R.
A Procedure for the Selection of Connectives in Text Generation.
Technical Report, Columbia University, New York, NY, 1989.
- [4] Smadja F.
Microcoding Lexicons with co-occurrence knowledge for language generation.
Technical Report CUCS-448-89, Columbia University, July, 1989.
- [5] Smadja F.
Dictionaries For Language Generation Accounting For Co-Occurrence Knowledge.
Technical Report CUCS-418-89, Columbia University, July, 1989.
- [6] Smadja F.
Computational Aspects Of Language Acquisition.
Technical Report CUCS-399-88, Columbia University, July, 1989.
- [7] Wolz, U.
Automated consulting for extending user expertise in interactive environments: A task centered approach.
Technical Report CUCS-393-88, Department of Computer Science, Columbia University, New York, NY, 1988.
- [8] Wolz, U.
Finding a better way: Choosing and explaining alternative plans.
Technical Report CUCS-409-88, Department of Computer Science, Columbia University, New York, NY, 1988.
- [9] Wolz, U. and G.E. Kaiser.
An Automated Consultant for Interactive Environments.
Technical Report CUCS-391-88, Department of Computer Science, Columbia University, New York, NY, 1988.

Presentations

"Language Generation at Columbia: the Radio Project," Kathleen McKeown, invited speaker, European Workshop on Language Generation, Univ. of Edinburgh, Edinburgh, Scotland, March 1989.

"Research in Natural Language Generation," Kathieen McKeown, IBM sponsored Industry Government Symposium for NSF Presidential Young Investigators, April 1989.

"Language Generation for Equipment Maintenance and Repair." Kathleen McKeown, GE, Schenectady, N.Y., July 1989.

"Acquiring Lexical Co-occurrence Knowledge from Large On-line Corpora," Frank Smadja, Bell Communications Research, Morristown, N.J., August 1989.

"Microcoding Co-occurrence Knowledge in the Lexicon for Language Generation" Frank Smadja, AT&T Bell Laboratories, Murray Hill, N.J., September 1989.

"Highlighting User Related Advice," AAAI Workshop on Explanation, AAAI, St. Paul, Minn., August 1988.

"Comparison of Surface Language Generators: A Case Study in Choice of Connectives," 4th International Language Generation Workshop, Santa Catalina Island, Ca., July 1988.

"Getting Computers to Respond in English," invited talk, Women in Science Lecture Series, Brookhaven National Laboratory, Brookhaven, N.Y., May 1988.

"Individual Differences in Human Computer Interaction," panel member, CHI 88, Washington D.C., May 1988.

"Constraints on Text Generation," invited talk, University of Waterloo, Waterloo, Canada, January 1987.

5 Honors and Awards

K. McKeown, ACL Executive Board, 1989-1990.

K. McKeown, elected officer, American Association of Artificial Intelligence, spring 1987.

K. McKeown, NSF Presidential Young Investigator Award, May 1985-1990.

K. McKeown, IEEE Conference on AI Applications, Program Committee, 1989.

K. McKeown, Chair, IJCAI natural language program committee, 1987.

K. McKeown, Editorial Board, Computational Linguistics Journal, Feb. 1985 - Feb. 1988;

K. McKeown, Panel chair and session organizer for panel on "Text Generation," NCC, 1985;

K. McKeown, SIGART secretary/treasurer 1983-1987.

Wolz, U., guest editor, *User Modeling and User-adapted Interaction: an International Journal*, special issue on text generation and user modeling.

6 Research Transitions and DoD Interactions

The formalism which we will use to generate sentences for GENIE and to select lexical collocations, the Functional Unification Formalism (FUF) has been made available to Univ. of Delaware (Kathleen McCoy), Univ. of Illinois (Ken Forbus), Paris University (Laurence Danlos), and we are currently involved in legal arrangements to make it available to Bellcore. One of my supported graduates students, Frank Smadja, has regular interaction with people at Bellcore who are quite interested in his work on EXTRACT. In addition, he has had discussions with Ken Church at AT&T Laboratories. It is quite likely that Bellcore will eventually use some of his work in their systems.

We have made a presentation to DARPA on related work on language generation being carried out under a DARPA contract. For example, work on FUF has been primarily supported by DARPA, but has received some support from ONR and will move into our ONR supported systems this fall (which means further development of the system). We will be attending the DARPA Speech and Natural Language Workshop in October 1989 and presenting a paper on our DARPA supported research.

7 Software and Hardware Prototypes

We have constructed three software prototypes. None of these have been commercialized. Neither do we have plans to commercialize them in the future. However, they do have potential for use as tools in natural language systems.

1. Functional Unification Formalism (FUF): This is a surface language generator that can produce English sentences given a content specification, represented as a logical form in FUF notation. As noted above, this system has been made available to other universities and we are currently undergoing the legal process to make it available to Bellcore. This system will be used in GENIE, our intelligent tutor. This work is partially supported by ONR and partially supported by DARPA.
2. We have developed a knowledge representation system for goals and plans (embedded in Sun's Hyperclass system) and are using it to represent commands, plans, and goals for a software development environment called MARVEL being developed by Prof. Gail Kaiser and her students.
3. We have developed a semantic interpreter which, given a meta-structure representation for an expert system rule base, can parse English sentences and questions in conjunction with an ATN interpreter. This has been tested in two domains and we are considering using it for a third large system under development for DARPA.
4. We have developed a question answering system, ADVISOR, which can answer questions courses and whether a student can or should take a specified course. Explanations are tailored to the goals and interests of the students, which are derived by tracking questions asked over the session. Input to ADVISOR is a natural language question. Output is a one to several sentence answer.