

# RESEARCH IN NATURAL LANGUAGE PROCESSING

## University of Pennsylvania Department of Computer and Information Science

This a brief report summarizing our work to date, our intermediate and long term goals, and a summary of some of our publications.

**FACULTY** Aravind Joshi, Tim Finin, Dale Miller, Lokendra Shastri, and Bonnie Webber  
**STUDENTS** Brant Cheikes, John Dowding, Amy Felty, Ellen Hays, Robert Kass, Ron Katriel, Sitaram Lanka, Megan Moser, Gopalan Nadathur, MaryAngela Papalaskaris, Martha Pollack, Robert Rubinoff, Yves Schabes, Ethel Schuster, Sunil Shende, Jill Smudski, Vijayshankar, David Weir, Blair Whitaker  
**FACILITIES** LINC (Language, Information, and Computation) laboratory, which consists of a dedicated VAX 11/785, 10 Symbolics Lisp machines, 7 HP 68020 based AI workstations, a SUN workstation, several Macintoshes, and a laser printer. These machines are networked together and to other research facilities in the department.

### MAJOR THRUST

Natural language interfaces providing support for many different communicative functions.

- Providing definitions of concepts
- Recognizing and correcting user misconceptions
- Providing explanations
- Offering to provide information later, when known
- Verifying and demonstrating understanding
- Exploiting and enriching the context of natural language discourse between user and system.

### WORK-TO-DATE

- Integration of RUS-TEXT-MUMBLE (RTM) - This effort involves integrating three natural language system components (BBN's RUS parser-interpreter, McKeown's TEXT system (developed at Penn), and McDonald's MUMBLE system (received from U. Mass in January 1985). This integration of three independently developed systems has required substantial effort. The version of RTM (to be completed in May 1986) [1] accepts a limited number of English language requests for definitions of, descriptions of, or comparisons between terms in the ONR database used by Kathy McKeown in her development of TEXT; [2] formulates appropriate responses using TEXT and outputs those responses in English using MUMBLE; and [3] runs on a SYMBOLICS Lisp machine. This work has been done by Moser, Whitaker and Rubinoff.
- Initial work on incorporating a sense of *relevance* in monitor offers. Mays' dissertation work on monitor offers was limited to issues of *competency*. This work is being done by Cheikes and Schabes.
- Completion of McCoy's dissertation work on correcting certain types of object-related misconceptions and implementation of a system called ROMPER which generates such corrections. (MUMBLE is used as the tactical generation component of this system as well.)
- Completion of Hirschberg's dissertation work on scalar implicatures and their use in constructing non-misleading responses.
- Completion of Pollack's dissertation work on plan inference in which user's and system's beliefs about actions and plans is decoupled.
- Continuation of work on integrating scalar-implicature-based reasoning within a general framework of circumscription-based non-monotonic reasoning.
- Development of methods for converting proofs in a system akin to first-order resolution into natural deduction (ND) proofs, which are then reorganized into cohesive paragraphs using Chester's 1976 algorithm.

- Development of methods of converting modal resolution proofs into modal ND proofs and higher-order resolution proofs into higher-order ND proofs.
- Initial development of domain-independent tools for expressing and reasoning about user models - in particular, for defining hierarchies of stereotypical users, representing individual users, and drawing inferences about them using a default logic.
- Continuation of basic research on local coherence of discourse using the notions of centering and syntax, semantics, and parsing of tree adjoining grammars.

## **FUTURE PLANS**

Having gained the experience of integrating three natural language systems and carrying out some of the basic research as described in the previous section, we have now developed the plan described below, which summarizes the near term and long term goals.

### **Near Term Goals**

We have three tangible goals for the next year:

- Completing the RTM demonstration system (using the existing domain and knowledge representation) and producing a videotape which explains and demonstrates it.
- Developing TEXT into a more modular tool for defining and comparing terms, on the order of RUS and MUMBLE. This will eliminate its tie to a particular knowledge representation and increase its portability.
- Acquiring familiarity with the PENMAN approach to NL generation through acting as a beta-test site for NIGEL.

### **Long Term Goals**

#### **Support for NL Definitions - Enriched Knowledge Representation**

In our original proposal, we stated our intention of employing a richer knowledge representation as the basis for our work on text generation, especially for constructing definitions. Our original idea was to make use of BBN's NIKL system. In the past year though, we have become aware of some of NIKL's limitations, which essentially make it non-optimal, even as a next step, for our text generation work. On the other hand, we have identified several features with which a NIKL-like language could be enriched to make it more suitable for our work:

- associating non-definitional information with concepts in a way that maintains the underlying structure of that information, without interfering with NIKL's automatic classification mechanism.
- associating "evidential" information with concepts, especially frequency information - how often the concept is known to display particular features.
- allowing for what appears to be conflicting information coming down through inheritance - e.g., information that is contrary to expectations grounded in an alternative perspective on a concept
- allowing mutual definition of concepts - each being defined with reference to the others in a set
- incorporating notions of time and change - allowing the defining properties and evidential properties of concepts to include how they change over time
- allowing assertions about usual relations between properties of subtypes

Work on an enriched knowledge representation that includes all these features in a well-motivated way will take several years. However one that includes at least the first three of them can probably be developed over the next two years, with work on employing it in text generation beginning after the first six months to a year after the start of that work.

#### **Support of NL Definitions - Use of Discourse and User Models**

The TEXT system, as it is currently structured, will produce the same definition for a concept (or comparison between two concepts) whenever it is asked. It does not take into account what the user may have already found out about the concept, or what it is implicitly being contrasted with (e.g., some other concept the user has recently asked about), or what the user's goal is in making the request. Hence, other directions in which we would like to take this definitional/clarificational capability is to increase its sensitivity to (1) the discourse history, to avoid repetition and possibly to take advantage of the additional clarity brought by contrasting a new term to one explained before; (2) the user's level of expertise, to avoid either stating the obvious or going more deeply into a concept than the user can understand; and (3) the user's goals, to focus on those aspects of the concept being defined (or concepts being compared) which are significant to the current task. (The latter is related to the notion of "perspective" used in Kathy McCoy's recent thesis here.) For both these aspects of user modelling (in contrast with the first point, which can be developed using the current discourse alone), we will draw on the other work being done here on domain-independent user-modelling mechanisms. This proposed work must be done in a domain in which tasks can be characterized and recognized. Thus we plan to do this initially in investment advising domain that we have started to develop. Work on

incorporating and using discourse history will involved about a one-year effort, once the knowledge base is built. Work on incorporating and using a model of a user's expertise and goals will take more time, on the order of two to three years.

## Explanations

Again in our original proposal, we proposed work on constructing natural language explanations - more specifically, on ways to loosen the current tight coupling between the form of the system's proof of some statement to the form of its explanation of why the statement is true. This coupling has kept systems which should be able to explain their reasoning from employing stronger proof methods which do not have a natural, understandable form of presentation to their human users.

Our immediate goals involve:

- developing a demonstration system which responds to NL queries posed to RUS by doing an efficient first-order resolution-based proof, transforming that proof into an ND proof, organizing that proof according to an improved version of the Chester algorithm, and then producing an English version of the text using MUMBLE or NIGEL.
- abstracting from the three separate sets of proof conversion methods (noted under WORK-TO-DATE) into general methods of transforming any resolution-style proof in any logic into its corresponding ND proof.
- determining whether existing methods of organizing first-order ND proofs into paragraphs are applicable to ND proofs in these stronger logics or whether more must be done to produce high-quality, cohesive, understandable text.

Our long-term goals remain as stated in our original proposal - the production of explanations sensitive to users' beliefs, expertise, desired level of detail and expectations. In this long-term research, we see taking expertise and desired level of detail into account in determining how much of the ND proof is made explicit. Of more interest is how users' beliefs and expectations should affect the explanations. Work on scientific explanation has shown that central to the explanation of what is the case is a set of alternative situations which are not the case. One explains what is in contrast to what is not. However, this requires additional work, to prove of each of the alternatives (which may be given explicitly by the user - "why this and not that?" - or inferred from the system's model of the user's expectations) that it is not true. Our planned approach involves guiding the (failing) proof of each alternative against the successful proof. The point is that although there may be many failing proofs of each alternative, the most relevant of these in the current situation is the one which is analogous - up to the point of failure - to the original successful proof not only should this technique provide relevant information, but it should also be efficient in reducing the search space. We expect this work to take on the order of two to three years, provided we have enough resources to pursue it in parallel with our more near-term goals.

## Natural Language Parsing and Generation

While continuing to use the RUS system, we will continue our work on tree adjoining grammar (TAG) both from the parsing and generation points of view. TAGs lead to some attractive approaches to parallelizing parsing and also seem to provide natural planning units for generation. This work will be integrated with our future work on parsing and generation. Our first language generator (used by TEXT) was one based on Kay's Functional Unification Grammar. While theoretically elegant, it was unacceptably slow (in its straightforward implementation), leading us last year to import the MUMBLE generator from McDonald at University of Massachusetts and adapt it to work with TEXT. Using MUMBLE has produced a 60-fold speed-up in generation time. However, adapting MUMBLE to work with TEXT and, independently, with two other systems has made us aware of MUMBLE's limitations, primarily its lack of knowledge of words or grammar. Essentially, MUMBLE's knowledge is limited to how to realize particular message units (i.e., to choose an acceptable one from an *a priori* specified set of choices), given constraints already imposed by message units that have already been realized. The large amount of work that must be invested in building a MUMBLE lexicon and the lack of inter-application portability of anything but the control structure comes from this fact - that one has to completely specify each set of choices beforehand for each message unit and *the sets are completely application specific*. We propose to work on the development of a new architecture, including our work on tags, that avoids these limitations by having more knowledge of syntax and words and hence is more portable between applications. The time frame for this project is approximately three years.

## Anaphora Resolution

The RUS parser/interpreter we received from BBN uses a limited method of resolving definite pronouns and noun phrases that is only a bit more advanced than the one originally developed for BBN's LUNAR system back in 1971. Since then, there have been major theoretical advances in our understanding of discourse anaphora (in the works of Grosz (at SRI), Joshi, Sidner (at BBN), Webber, and Weinstein), but these theoretical advances have not yet found their way into natural language understanding systems. We feel strongly qualified to undertake this work, having two of the major participants (Joshi and Webber) here at Penn already, and want to do so. For us, it is both of research interest and of practical importance, since it can mean a major improvement in system's understanding abilities. We will also integrate our work on tags with this effort as it relates to parsing and generation. This work will also complement additional work being done here on a theoretical and computational account of anaphoric reference to actions and events. We see this work as taking about two to two and a half years.

## User Modeling

The need for systems to model the knowledge and beliefs of their users has already been pointed out. We plan to address a number of issues which underly the successful development and incorporation of explicit user models. Our current domain-independent user-modelling system, GUMS, provides mechanisms for defining hierarchies of stereotypical users, representing individual users, and drawing inferences about them using a rich default logic. We will continue to develop this system as a tool which will support the user modeling needs of various applications. We also plan to study the problem of how new knowledge of individual users can be derived from their regular interaction - that is, how relevant information about users can be inferred from their queries and responses. In other situations it may become necessary for the system to explicitly pose a few crucial questions to the user to determine what he or she does and does not know.

## System Integration

Finally, we plan to begin work on system integration. In recent years, we have identified many types of behavior that interfaces to database systems and expert systems should demonstrate. Beginning with Kaplan's work on recognizing and responding to existential presupposition failures in his COOP system, we have developed and produced several modules, each demonstrating another type of desired behavior. These include the ability to recognize and respond to type failures, the ability to respond to object-related misconceptions, the ability to calculate and offer competent database monitors, the ability to use scalar implicatures to convey additional information, and the ability to respond to a class of "inappropriate" queries, and various paraphrase abilities.

Following the publication of Kaplan's thesis, the features of his COOP system were soon incorporated into several database interfaces (both natural slanguage and formal query language). This gave the resulting systems the ability to give two types of responses: either a direct answer, if there was one, or a statement concerning the absence of individuals satisfying some description in the given query. Now we plan to tackle the more significant problem related to this:

Given a system that is able to call upon a variety of response strategies, how does it decide what to do in a given circumstance? This is the issue we plan to explore by investigating the integration of multiple communicative behaviors. Given a system with several different types of useful behaviors, which can be combined in various ways, can one efficiently and effectively coordinate a response that is better (i.e., more useful, more helpful and more understandable) than simply a (direct) answer. While we speculate that it will be the case that identifying what one might consider the *best* response might take complex reasoning about the user's goals, level of expertise and need-to-know with respect to what the answer (if any) actually is, we also plan to look at how, with more limited resources, we can still improve system behavior.

This aspect of our future plans is the most long term, involving both the actual component integration itself (in which, in many cases, it is only the basic ideas that can be carried over, where the component must be re-programmed entirely to fit into the integrated system) and the development of that part of the total system that reasons about what kind of response(s) to give. The time frame here is approximately four years.

## Architecture

We plan to investigate parallel and connectionist architectures and algorithms for realizing our systems, especially those for knowledge representation, reasoning, explanations, and integrated parsing and generation.

## Abstracts of Recent Technical Reports

***INTERACTIVE CLASSIFICATION A Technique for the Aquisition and Maintenance of Knowledge Bases*, Tim Finin and David Silverman, MS-CIS-84-17.**

The practical application of frame-based knowledge-based systems, such as in expert systems, requires the maintenance of potentially very large amounts of declarative knowledge stored in their knowledge bases (KBs). As a KB grows in size and complexity, it becomes more difficult to maintain and extend. Even someone who is familiar with the representation and the contents of the existing KB may introduce inconsistencies and errors whenever an addition or modification is made.

This paper describes an approach to this problem based on a tool called an interactive classifier. An interactive classifier uses the contents of the existing KB and knowledge about its representation to assist the person who is maintaining the KB in describing new KB objects. The interactive classifier will identify the appropriate taxonomic location for the newly described object and add it to the KB. The new object is allowed to be a generalization of existing KB objects, enabling the system to learn more about existing objects. The ideas have been tested in a system call KuBIC, for Knowledge Base Interactive Classifier, and are being extended to a more complete knowledge representation language.

***Correcting Object-Related Misconceptions: How Should The System Respond?*, Kathleen F. McCoy, MS-CIS-84-18.**

This paper describes a computational method for correcting users' misconceptions concerning the objects modeled by a computer system. The method involves classifying object-related misconceptions according to the knowledge-base feature involved in the incorrect information. For each resulting class sub-types are identified, according to the structure of the knowledge base, which indicate what information may be supporting the misconception and, therefore, what information to include in the response. Such a characterization, along with a model of what the user knows, enables the system to reason in a domain-independent way about how best to correct the user.

***Default Reasoning in Interaction*, Aravind Joshi, Bonnie Webber, and Ralph Welschedel, MS-CIS-84-58**

Nonmonotonic reasoning is usually studied in the context of a logical system in its own right or as reasoning done by an agent, in which the agent reasons about the world from partial information and hence may draw conclusions unsupported by traditional logic. The main point of departure here is looking at nonmonotonic reasoning in the context of interacting with another agent. This information is partial, in that the other agent neither will not can make everything explicit. Knowing this, the agent may attempt to derive more from the interaction than what has been made explicit, by reasoning by default about what has been made explicit (often by contrast with what he assumes would have been made explicit, were something else the case). Thus there can be rules for default reasoning that are operative in the interactive situation ("interactional defaults") that are not operative with only a single agent.

***Preventing False Inferences*, Aravind Joshi, Bonnie Webber, and Ralph M. Welschedel, MS-CIS-84-59**

In cooperative man-machine interaction, it is taken as necessary that a system truthfully and informatively respond to a user's question. It is not, however, sufficient. In particular, if the system has reason to believe that its planned response might lead the user to draw an inference that it knows to be false, then it must block it by modifying or adding to its response. The problem is that a system neither can nor should explore all conclusions a user might possibly draw: its reasoning must be constrained in some systematic and well motivated way.

***Living Up To Expectations: Computing Expert Responses* Aravind Joshi, Bonnie Webber, and Ralph Welschedel, MS-CIS-84-60**

In cooperative man-machine interaction, it is necessary but not sufficient for a system to respond truthfully and informatively to a user's question. In particular, if the system has reason to believe that its planned response might mislead the user, then it must block that conclusion by modifying its response. This paper focusses on identifying and avoiding potentially misleading responses by acknowledging types of "informing behavior" usually expected of an expert. We attempt to give a formal account of several types of assertions that should be included in response to questions concerning the achievement of some goal (in addition to the simple answer), lest the questioner otherwise be misled.

***A Modal Temporal Logic for Reasoning About Changing Databases with Applications to Natural Language Question Answering*, Eric Mays, Aravind Joshi, Bonnie Webber, MS-CIS-85-01.**

A database which models a changing world must evolve in correspondence to the world. Previous work on natural language question answering systems for databases has largely ignored the issues which arise when the database is viewed as a dynamic (rather than a static) object. We investigate the question answering behaviors that become possible with the ability to represent and reason about the possible evolution of a database. These behaviors include offering to monitor for a possible future state of the database as an indirect response to a query, and directly answering questions about prior and future possibility. We apply a propositional modal temporal logic that captures possibility and temporality to represent and reason about dynamic databases, and present a sound axiomatization and proof and proof procedure.

***Explaining Concepts in Expert Systems: The CLEAR System*, Robert Rubnoff, MS-CIS-85-06, LINC LAB 02**

Existing expert systems provide limited explanatory ability. They can explain the specific reasoning the system uses, but if the user is confused about the concepts and terms the system is using, no help is available. The CLEAR system allows users to ask for explanations of specific concepts. The system generates the explanations by examining the rule base, selecting rules that are relevant to the concept asked about. These rules are then turned into English by various simple translation schemes and presented to the user, providing an explanation of how the concept is used by the system.

***The Linguistic Relevance of Tree Adjoining Grammars*, Anthony S. Kroch, Department of Linguistics, and Aravind K. Joshi, Department of Computer and Information Science, MS-CIS-85-16, LINC LAB 03**

In this paper the linguistic significance of the Tree Adjoining Grammar (TAG) has been investigated. An important property of TAG is that it defines a constrained theory of syntactic embedding, one requiring that embedded structures be composed out of elementary structures in a fixed way, and one which forces co-occurrence relations between elements that are separated in surface constituent structures to be stated broadly as constraints on elementary trees in which those elements are copresent. The extra generative power of TAG beyond context-free grammar emerges as a corollary of factoring recursion and co-occurrence relations. The linguistic details specifically discussed are raising constructions, passive, and WH-movements.

***A Computational Logic Approach to Syntax and Semantics* Dale A. Miller and Gopalan Nadathur, MS-CIS-85-17**

It is well known that higher-order logics are very expensive, and for this reason have been used to represent many problems in mathematics and theoretical computer science. In the latter domain, higher-order logics are often used to describe the semantics of first-order logics, natural languages, or programs, since the formalization of such semantics needs a recourse to quantification over the domain of functions and sets. In these settings, higher-order logic has generally been limited to a descriptive role. Once the formalization is made little has been made of it computationally, largely because there is abundant evidence that theorem proving in higher-order logics is very difficult. In this paper we shall look at a sublogic of a particular higher-order logic that is derived from Church's Theory of Types, and examine its representational power and its computational tractability. This sublogic can also be described as Horn clauses logic extended with quantifications over function variables and  $\lambda$ -contraction. We shall present a sound and complete theorem prover for this logic, which uses higher-order unification and may be described as an extension of a unification procedure for the typed  $\lambda$ -calculus. There are at least three ways in which this logic is different from the first-order logic that it generalizes. First it possesses function variables which can be instantiated with  $\lambda$ -terms and evaluated through  $\lambda$ -contractions. This provides the logic with a new source of computation. Second, since  $\lambda$ -terms do not have most general unifiers, the process of finding appropriate unifiers must branch, and hence involves real search. This facet provides a new source of nondeterminism in specifying computations. Finally, this logic can directly encode first-order logic in its term structure and can manipulate such terms in logically meaningful ways. We illustrate this with examples taken from knowledge representation and natural language parsing.

***The Role of Perspective In Responding to Property Misconceptions*, Kathleen F. McCoy, MS-CIS-85-31, May 1985**

In order to adequately respond to misconceptions involving an object's properties, we must have a context-sensitive method for determining object similarity. Such a method is introduced here. Some of the necessary contextual information is captured by a new notion of *object perspective*. It is shown how object perspective can be used to account for different responses to a given

misconception in different contexts.

***Some Computational Properties of Tree Adjoining Grammars*, Vijayshenkar and Joshi, MS-CIS-85-07**

Tree Adjoining Grammar (TAG) is a formalism for natural language grammars. Some of the basic notions of TAG's were introduced in [Joshi, Levy, and Takahashi 1975] and by [Joshi, 1983]. A detailed investigation of the linguistic relevance of TAG's has been carried out in [Kroch and Joshi, 1985]. In this paper, we will describe some new results for TAG's, especially in the following areas: (1) parsing complexity of TAG's, (2) some closure results for TAG's, and (3) the relationship to Head grammars.

***Grammar, Phrase Structure*, Aravind K. Joshi, MS-CIS-85-45**

Phrase-structure trees (phrase-markers) provide structural descriptions for sentences. Phrase-structure trees can be generated by phrase-structure grammars. Phrase-structure trees can be shown to be appropriate to characterize structural descriptions for sentences, including those aspects which are usually characterized by transformational grammars, by making certain amendments to CFG's, without increasing their power, or by generating them from elementary trees (phrase-markers) by a suitable rule of composition, increasing the power only mildly beyond that of CFG's. Structural descriptions provided by phrase-structure trees are used explicitly or implicitly in natural language processing systems.

***Question, Answer and Responses: Interacting with Knowledge Base Systems*, Bonnie Lynn Webber, MS-CIS-85-50, LINC LAB 04**

The purpose of this chapter is to examine the character of information-seeking interactions between a user and a knowledge base system (KBS). In doing so, I advocate that a clear distinction be made between an answer to a question and a response. The chapter characterizes questions, answers, and responses, the role they play in effective information interchanges, and what is involved in facilitating such interactions between user and KBS.

***A Theory Of Scalar Implicature*, Julia Bell Hirschberg, MS-CIS-85-56.**

***The Relationship Between Tree Adjoining Grammars And Head Grammars*, K. Vijay-Shanker, David J. Welr and Aravind K. Joshi, MS-CIS-86-01, LINC LAB 06**

Tree Adjoining Grammars (TAG) and Head Grammars (HG) were introduced to capture certain structural properties of natural languages. These formalisms, which were developed independently, appear to be quite different notationally. In this paper we discuss the formal relationship between the class of languages generated by TAG's (TAL) and the class of languages generated by HG's (HL). In particular, we show that HL's are included in TAL's and that TAG's are equivalent to a modification of HG's called Modified Head Grammars (MHG's). The inclusion of MHL in HL, and thus the equivalence of HG's and TAG's, in the most general case remains to be established. We show that this relationship is very close both linguistically and formally, the difference hinging on the status of heads of empty strings and whether one deals with heads directly or with the left and right wrapping positions around the head.

***Natural Language Interactions With Artificial Experts*, Tim Flnn, Aravind K. Joshi and Bonnie Lynn Webber, MS-CIS-86-16, LINC LAB 08).**

The aim of this paper is to justify why Natural Language (NL) interaction, of a very rich functionality, is critical to the effective use of Expert Systems and to describe what is needed and what has been done to support such interaction. Interactive functions discussed here include defining terms, paraphrasing, correcting misconceptions, avoiding misconceptions and modifying questions.

***Higher-Order Logic Programming*, Dale A. Miller and Gopalan Nadathur, MS-CIS-86-17**

In this paper we consider the problem of extending Prolog to include predicate and function variables and typed  $\lambda$ -terms. For this purpose, we use a higher-order logic to describe a generalization to first-order Horn clauses. We show that this extension possesses certain desirable computational properties. Specifically, we show that the familiar operational and least fixpoint semantics can be given to these clauses. A language,  $\lambda$ Prolog that is based on this generalization is then presented, and several examples of its use are provided. We also discuss an interpreter for this language in which new sources of branching and backtracking must be accommodated. An experimental interpreter has been constructed for the language, and all the examples in this paper have been tested using it.

***Some Uses of Higher-Order Logic in Computational Linguistics*, Dale A. Miller and Gopalan Nadathur, MS-CIS-86-31, LINC LAB 08**

Consideration of the question of meaning in the framework of linguistics often requires and allusion to sets and other higher-order notions. The traditional approach to representing and reasoning about meaning in a computational setting has been to use knowledge representation systems that are either based on first-order logic or that use mechanisms whose formal justifications are to be provided after the fact. In this paper we shall consider the use of a higher-order logic for this task. We first present a version of definite clauses (positive Horn clauses) that is based on this logic. Predicate and function variables may occur in such clauses the terms in the language are the typed  $\lambda$ -terms. Such term structures have a richness that may be exploited in representing meanings. We also describe a higher-order logic programming language, called  $\lambda$ Prolog, which represents programs as higher-order definite clauses and interprets them using a depth-first interpreter. A virtue of this language is that it is possible to write programs in it that integrate syntactic and semantic analyses into one computational paradigm. This is to be contrasted with the more common practice of using two entirely different computation paradigms, such as DCGs or ATNs for parsing and frames or semantic nets for semantic processing. We illustrate such and integration in this language by considering a simple example, and we claim that its use makes the task of providing formal justifications for the computations specified much more direct.

***Some Aspects Of Default Reasoning In Interactive Discourse*, Aravind K. Joshi, Bonnie L. Webber and Ralph M. Welschedel, MS-CIS-86-27 (revised version of MS-CIS-84-58)**

In cooperative interaction, it is taken as necessary that a system truthfully and informatively respond to a user's question. It is not, however, sufficient. In particular, if the system has reason to believe that its planned response might lead the user to draw an inference that it knows to be false, then it must block it by modifying or adding to its response. In this paper we investigate several aspects of such reasoning in interactive discourse.

***Adapting MUMBLE: Experience with Natural Language Generation*, Robert Rubinoff, MS-CIS-86-32, LINC LAB 09}**

This paper describes the construction of a MUMBLE-based [McDonald 83b tactical component for the TEXT text generation system [McKeown 85]. This new component, which produces fluent English sentences from the sequence of structured message units output from TEXT's strategic component, has produced a 60-fold speed-up in sentence production. Adapting MUMBLE required work on each of the three parts of the MUMBLE framework: the interpreter, the grammar, and the dictionary. It also provided some insight into the generation process and the consequences of MUMBLE's commitment to a deterministic model.

***GUMS<sub>1</sub> : A General User Modeling System*, Tim Finin and David Drager, MS-CIS-86-35**

This paper describes a general architecture of a domain independent system for building and maintaining *long term models of individual users*. The user modeling system is intended to provide a well defined set of services for an *application system* which is interacting with various users and has a need to build and maintain models of them. As the application system interacts with a user, it can acquire knowledge of him and pass that knowledge on to the user model maintenance system for incorporation. We describe a prototype *general user modeling system* which we have implemented in Prolog. This system satisfies some of the desirable characteristics we discuss.

***Breaking the Primitive Concept Barrier*, Robert Kass, Ron Katriel, and Tim Finin, MS-CIS-86-36**

Building and maintaining a large knowledge base of general information requires a knowledge representation system with precise semantics and an easy knowledge acquisition procedure. Systems such as KL-ONE meet these criteria by using a classifier to install new concepts into a taxonomic structure. These systems use a formal notion of a definition for concepts. Unfortunately, many concepts do not seem to have such precise definitions, and end up represented as primitive concepts. Primitive concepts form a barrier to classification, forcing the user to manually classify a new concept with respect to all primitive concepts in the knowledge base.

We propose an extension to KL-ONE which retains its soundness and greatly reduces the burden on the user during knowledge acquisition. This extension consists of adding an explicit definitional component to concepts and relaxing the strictness of concept definitions themselves. The relaxed definition reduces the number primitive concepts in a knowledge base, enables the classifier to handle concepts that do not have complete definitions and enhances the usefulness of an interactive classifier.