KA: Integrating Natural Language Processing and Problem Solving *

Jeff Pittges, Kurt Eiselt, Ashok Goel,
Andrés Gómez de Silva Garza, Kavi Mahesh, and **Justin Peterson**

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
ka@pravda.cc.gatech.edu

Abstract

Traditional Cognitive Science has studied various cognitive components in isolation. Our project attempts to alleviate some of the problems with this separation by focusing on the role of problem solving in language comprehension. Specifically, the KA project integrates six areas of current investigation in Cognitive Science: knowledge representation, memory organization, language comprehension, knowledge acquisition, problem solving, and control architectures. We are developing a ***model-based text interpretation and knowledge acquisition system*** which, when completed, will be able to read and interpret descriptions of physical devices, construct models of the devices, and use the acquired models to solve novel design problems. This paper presents three areas in which we use problem solving to constrain natural language understanding: (1) the use of mental models as a foundation for both problem solving and natural language understanding, (2) the use of design experience to influence the understanding process, and (3) the use of the design process to establish the cost of linguistic decisions.

Goals and Motivations

The cognitive abilities which comprise "human intelligence" are surely more than a collection of independent functions (e.g., perception, learning, problem solving, language comprehension). They are a set of faculties which combine to form a tightly integrated system in which each component relies on others in order to function most effectively. Traditional Cognitive Science, however, has primarily studied them in isolation, without regard to how the individual components may use the knowledge or processes commonly associated with other components. For example, much of the research on problem solving has focused on knowledge-based methods for solving complex problems without regard to the perceptual or linguistic processes involved. Similarly, work on natural language understanding typically views the language understander as an entity unto itself and ignores how language is used. We believe that robust theories will only emerge by studying several areas together, thereby sufficiently constraining our theories.

Our research attempts to alleviate the artificial separation between cognitive components by integrating problem solving and language understanding in terms of the knowledge representations and reasoning methods they use. Problem solving plays a crucial role in understanding natural language. Specifically, understanding requirement specifications written in natural language requires a problem solving context in which reasoning decisions can be linguistic decisions. Our work explores how the introduction of a problem solving task changes the linguistic problems encountered by the understander.

Our theory contends that (1) the use of mental models in the task of design forms a well-defined target space for linguistic output, (2) design experience provides exemplar interpretations which greatly influence the understanding process, and (3) the design process establishes the utility (cost) of linguistic decisions. The KA project is currently addressing three issues: ambiguity resolution, compensation for underspecification, and the removal of irrelevant details from consideration. We predict that (1) people prefer familiar interpretations even though alternative interpretations are plausible, (2) people select interpretations which entail the least reasoning effort, and (3) people pass over irrelevant information and extract only the details necessary to complete their mental models.

---

*This work has been supported by research grants from the National Science Foundation, the Office of Naval Research (contract N00014-92-J-1234), Northern Telecom, Georgia Tech Research Corporation, and computer equipment and software grants and donations from IBM, Symbolics, and NCR.

The Communication Gap

Research in natural language understanding has created a number of potentially artificial tasks (e.g., syntactic parsing) because the work has been performed in isolation. Real world tasks provide a context which refocuses many of the linguistic problems that have been central to the field.

Nevertheless, the general problem still remains: written language produces a communication gap because the author of a text often provides some of the details and expects the reader to fill in the rest. The author's omissions are typically unintentional and result from the inherent difficulties of communicating in natural language. This problem is further complicated when dealing with requirement specifications. Three characteristics of this general problem are of particular interest: (1) specifications written in natural language are incomplete, (2) the well-formed structures that language provides are ambiguous in the target domain, and (3) irrelevant information is given. The KA project addresses the following specific problems.

1. Ambiguity may occur at several levels. Therefore, requirement specifications written in natural language may specify a set of devices, rather than a single, unambiguous device, because ambiguity allows for several interpretations.
2. Natural language may underspecify the device to be designed. Coherent understanding requires inferring a significant amount of information which is left unspecified by the natural language surface form.
3. Natural language specifications contain irrelevant information which can detract from a clear understanding of what is to be provided by the device being described. Consequently, irrelevant details can lead to inefficiencies in the design process.

The Task for KA

The long term goal of our work is to develop a *model-based text interpretation and design system* called KA [Goel and Eiselt, 1991], which accepts a requirement specification written in English and produces a design which meets the specification. The design is expressed as a structure-behavior-function (SBF) model that specifies how the structure achieves its function [Chandrasekaran et al., 1993]. This paper focuses on the problem of interpreting input specifications.

Figure 1 shows an oversimplified *input* specification which illustrates some of the problems addressed by the KA project.

This specification is lexically ambiguous. The word "input" is used to describe an external stimulus (i.e., a small force on the switch) rather than some entity (e.g., a substance like electricity) which is consumed by the device. Linguistic knowledge alone is insufficient to resolve such ambiguity. In the context of a design task, however, a problem solving reasoner, using its knowledge of devices, could determine that "a small force" is an external stimulus. Specifically, the use of mental models based on an ontology of devices effortlessly guides the understander to the correct interpretation. Therefore, the target space of SBF device models, the design process, and previous design experience all contribute to resolve ambiguity.

Consider a flashlight circuit. The function of the circuit is to produce light. The input is a small force on the switch. The output is light of eighteen lumens intensity and blue color.

Figure 1: Example Requirement Specification

The example above is incomplete because there is no mention of how the device is to be powered. Therefore, the description actually specifies a design which uses batteries as well as a design which plugs into an outlet. Given a rich set of exemplars, however, it is possible to exploit previous experience and select a familiar interpretation (namely, flashlights use batteries). Furthermore, should simulation of the desired device produce inconsistencies, the interpretation can be modified. Such evaluation is not possible without problem solving capabilities.

Some of the information in the example text is irrelevant and/or redundant. Stating that the function of the circuit is to produce light is redundant given the more complete specification of the output which follows further in the text. Such information can be filtered from consideration if the mental models being used do not address these details.

The output of KA is a design expressed as an SBF model. As mentioned above, we are interested here only in the interpretation of the design specification, not in design problem solving. KA's interpretation of the example specification is a functional description shown in Figure 4.

In order to map requirement specifications to functional descriptions, KA applies both linguistic and problem solving knowledge to mutually constrain the understanding process. This effectively resolves ambiguity, fills in missing details, and filters irrelevant information from consideration. In this way, the KA system provides a robust context in which to effectively communicate in natural language.

KA Architecture

The architecture of KA is shown in Figure 2. The Capsulizer, Proposer, Filter, and Semantic Network have been modified from a natural language understander called ATLAST [Eiselt, 1989]. The Retriever, Adapter, Storer, and Case Memory have been modified from a problem solving design system called KRITIK [Goel, 1991].

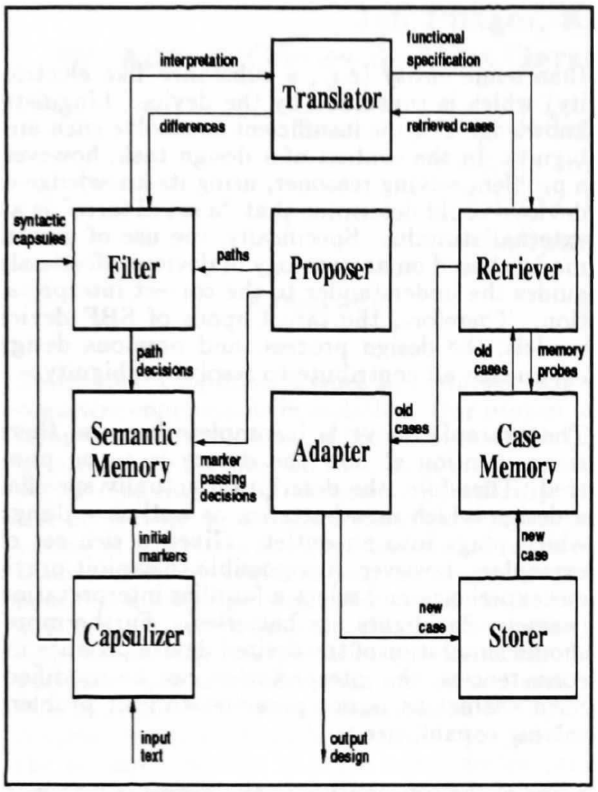

Figure 2: KA System Architecture

ATLAST

Eiselt's ATLAST language understanding model [Eiselt, 1989] is shown on the left side of Figure 2. The Capsulizer is an Augmented Transition Network (ATN) parser which produces syntactic structures. These units are sent to the Filter and initial markers are placed in the semantic network (see Charniak, Hendler, and Norvig [Charniak, 1981; Hendler, 1986; Norvig, 1989] for a discussion of marker passing in semantic networks). The Proposer generates paths through the network and the Filter uses a set of metrics (heuristics) to select an interpretation; a subset of paths from those which have been proposed.

The semantic network allows ATLAST to make inferences and provides the medium for KA to integrate linguistic and design knowledge. ATLAST has been designed to promote error recovery which allows KA to construct alternative interpretations when KRITIK identifies problems with ATLAST's output.

KRITIK

Goel's KRITIK [Goel, 1991] is a problem solver capable of designing devices. The system is based on mental models—topographic models of physical spaces, structure-behavior-function (SBF) models of physical devices, and causal models of physical processes. KRITIK integrates model-based reasoning with case-based reasoning by grouping the various models of a device into a case. Cases are indexed by a functional specification of the device [Goel, 1992].

The right side of Figure 2 shows the functional components of KRITIK. Given a functional specification of a desired design, the Retriever returns a set of cases which at least partially match the specification. These cases are given to the Adapter which modifies the retrieved design to produce the desired design. The Storer saves this design in the case memory thus acquiring a model of the new device for later reuse.

The KA System

Both ATLAST and KRITIK have been modified for the KA system so that they can communicate with each other; the two are no longer stand-alone systems. Because the two systems use different forms of representation (ATLAST uses a semantic network and KRITIK uses frames), ATLAST has been modified to produce its output in the form of frames and KRITIK has been modified to produce networks. These changes have been achieved by using a translator (shown in Figure 2) which performs a formal translation without affecting the content of the representations.

Control in the KA system is shared between ATLAST and KRITIK. In addition, ATLAST's original semantic network based on an ontology of events and actors has been replaced by KRITIK's ontology of physical devices. As the system develops, we expect the translator to gradually disappear and the KA system to evolve into a seamless, tightly integrated system.

In the current version of KA, the semantic network reflects KRITIK's ontology of devices (e.g., structure, behavior, function, substances, components, fields). Figure 3 shows a portion of the network. This knowledge allows KA to resolve ambiguities and filter irrelevant details. KRITIK contributes to the understanding process through its knowledge of devices, which is grounded in the target domain, but this does not prevent KA from constructing all possible interpretations. Those which fit best with KRITIK's ontology are selected first, but others may be chosen later depending on *feedback*

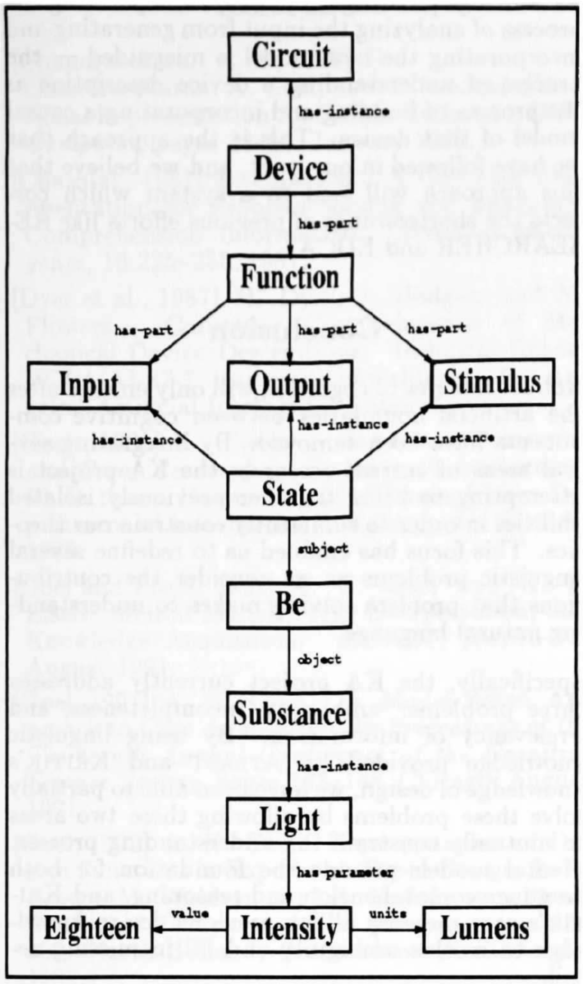

Figure 3: Partial Semantic Network in KA

from the Retriever. KA is able to do this by using ATLAST's error recovery capabilities. Therefore, KRITIK adds content and biases for understanding devices without limiting the understander.

KA Processing

The KA system provides a partial solution to the three problems discussed in section 2: ambiguity, incompleteness, and irrelevancy of information. In short, ambiguity is resolved using KRITIK's ontology along with feedback from the design process, incomplete specifications are filled in by recalling previous designs from KRITIK's memory, and irrelevant information is removed by applying KRITIK's ontology with ATLAST's metrics.

Using a set of six metrics (heuristics used to determine the "goodness" of a path), the Filter evaluates the paths identified by the Proposer. The paths which are selected by the Filter constitute KA's interpretation of the text. These paths are given to the Translator which produces a functional specification to be used as a memory probe by the Retriever. Figure 4 shows the functional specification for the example text shown in Figure 1. This translation is performed by traversing the paths of the interpretation and filling in the slots of a frame.

GIVEN: **NIL**

MAKES: **SUBSTANCE: Light**
Color: blue
Intensity: 18 lumens

Figure 4: KRITIK Functional Specification

The Retriever searches KA's case memory and retrieves a set of cases which at least partially match the given functional specification. The differences between the desired specification and those of the retrieved cases are returned to the Translator which uses the differences to add new markers to the semantic network. The Proposer and Filter use the new information to produce an alternative interpretation which is given to the Translator and the cycle repeats. The retrieved cases are given to the Adapter which tries to modify them to meet the desired specification. Once the Adapter constructs a model of the desired device, the Storer saves the model as a new case.

Related Work

Language understanding systems have traditionally used knowledge structures to guide the understanding process (e.g., Schank and Abelson's scripts [Schank and Abelson, 1977]). In addition, several attempts have been made to integrate natural language and problem solving using a common representation for both language comprehension and reasoning [Rieger, 1976; Simon and Hayes, 1979; Charniak, 1981; Wong, 1981]. Our work continues in this direction by applying KRITIK's content theory to the understanding process. The same mental models are used for comprehension and to reason about devices. In addition, the project focuses on redefining traditional linguistic problems based on the hypothesis that problem solving contributes to language understanding.

This research is inspired in part by Winograd's SHRDLU system [Winograd, 1972]. SHRDLU formed plans for actions in a simulated blocks world based on its interpretation of external commands expressed in English. It explored certain interactions between language understanding and planning, and demonstrated the methodological usefulness of exploiting the constraints imposed by planning on language understanding, and vice versa. Of course, SHRDLU also suffered from a number of well-known problems. For example, it assumed a closed world, it represented knowledge procedurally, it lacked the capability of abstract reasoning, and it also lacked sufficient control over processing.

The PROTEUS system [Ksiezyk and Grishman, 1989], comprehends device failure reports. PROTEUS, however, did not implement diagnosis and repair. Also, language understanding in PROTEUS is mainly driven by stored templates. KA uses a more general theory of language, embodied in ATLAST, and also does problem solving via KRITIK.

Lebowitz's RESEARCHER [Lebowitz, 1983] reads natural-language texts in the form of patent abstracts, specifically disk drive patents, and updates its long-term memory with generalizations made from these texts. RESEARCHER stores a generalized representation in its memory consisting of a *topographic model* of the disk drive. RESEARCHER's emphasis on components and topographic relationships leaves it unable to build causal models of the mechanisms described. In other words, RESEARCHER effectively knows how a disk drive is constructed, but not how it works. KA explicitly uses SBF models that capture teleological and causal relationships in addition to topographical ones. In addition, along with mental models, KA uses KRITIK's episodic knowledge of particular design cases.

Dyer, Hodges, and Flowers [Dyer et al., 1987] describe EDCA, a conceptual analyzer which serves as a natural language front-end for EDISON, a design problem solver. EDCA uses knowledge of the function of physical devices to produce an episodic description of a device's behavior as described by an input text. This episodic description can then be used to generate a new device model to be integrated into long-term memory. The result is a much more comprehensive understanding of the device's functionality than was possible with RESEARCHER, but EDCA's analysis of the device description is not fully integrated with the processes for generating new device models and incorporating them into memory. EDCA, in other words, is but a front end to EDISON.

As Selfridge [Selfridge, 1989] notes, separating the process of analyzing the input from generating and incorporating the new model is misguided — the process of understanding a device description *is* the process of building and incorporating a causal model of that device. This is the approach that we have followed in our work, and we believe that this approach will lead to a system which corrects the shortcomings of previous efforts like RESEARCHER and EDCA.

Conclusion

Robust theories of cognition will only emerge after the artificial boundaries between cognitive components have been removed. By integrating several areas of current research, the KA project is attempting to bring together previously isolated abilities in order to sufficiently constrain our theories. This focus has enabled us to redefine several linguistic problems as we consider the contributions that problem solving makes to understanding natural language.

Specifically, the KA project currently addresses three problems: ambiguity, incompleteness, and irrelevancy of information. By using linguistic knowledge provided by ATLAST and KRITIK's knowledge of design, we have been able to partially solve these problems by allowing these two areas to mutually constrain the understanding process. Mental models provide the foundation for both language comprehension and reasoning, and KRITIK's case memory allows previous design knowledge to resolve ambiguity and fill in missing details.

This work is a natural extension of our earlier research on model-based design problem solving and device knowledge acquisition [Goel, 1992]. When completed, the new system will be able to read and interpret device descriptions, construct SBF models of the devices, and use the acquired models in solving novel design problems [Goel and Eiselt, 1991].

Acknowledgements

We thank the other members of the KA project, Sambasiva Bhatta and Eleni Stroulia, for their contributions to KRITIK, and Frédérique Bonatto for her help with ATLAST. This work has been generously supported by a research grant from Northern Telecom. It has also been supported by grants from the National Science Foundation, the Office of Naval Research (contract N00014-92-J-1234), the Georgia Tech Research Corporation, and computer equipment and software grants and donations from IBM and Symbolics. In addition, Jeff Pittges is supported by a graduate fellowship from BNR. We are grateful to Robert Bloeden, David May, Dick Peacocke and Deborah Stokes for their support and encouragement of this work.

References

[Chandrasekaran et al., 1993] B. Chandrasekaran, A. Goel, and Y. Iwasaki. Functional Representation As Design Rationale. *IEEE Computer*, pages 48–56, January 1993.

[Charniak, 1981] E. Charniak. A Common Representation for Problem-Solving and Language-Comprehension Information. *Artificial Intelligence*, 16:225–255, 1981.

[Dyer et al., 1987] M. Dyer, J. Hodges, and M. Flowers. Computer Comprehension of Mechanical Device Descriptions. Technical Report UCLA-AI-87-7, University of California, Los Angeles, Artificial Intelligence Laboratory, 1987.

[Eiselt, 1989] K. Eiselt. *Inference Processing and Error Recovery in Sentence Understanding.* Ph.D. thesis, University of California, Irvine, Department of Information and Computer Science, 1989.

[Goel and Eiselt, 1991] A. K. Goel and K. P. Eiselt. Mental Models, Text Interpretation, and Knowledge Acquisition. *SIGART*, 2(4):75–78, August 1991.

[Goel, 1991] A. K. Goel. A Model-Based Approach to Case Adaptation. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 143–148, Chicago, August 1991.

[Goel, 1992] A. K. Goel. Representation of Design Functions in Experience-Based Design. In D. Brown, M. Waldron, and H. Yoshikawa, editors, *Intelligent Computer Aided Design*, pages 283–308. North-Holland, Amsterdam, Netherlands, 1992.

[Hendler, 1986] J. A. Hendler. Integrating Marker-Passing and Problem-Solving: A Spreading Activation Approach to Improved Choice in Planning. Technical Report CS-86-01, Brown University, Department of Computer Science, 1986.

[Ksiezyk and Grishman, 1989] T. Ksiezyk and R. Grishman. Equipment Simulation for Language Understanding. *International Journal of Expert Systems*, 2(1):33–78, 1989.

[Lebowitz, 1983] M. Lebowitz. RESEARCHER: An Overview. In *Proc. National Conference on Artificial Intelligence*, pages 232–235, 1983.

[Norvig, 1989] P. Norvig. Marker Passing as a Weak Method for Text Inferencing. *Cognitive Science*, 13:569–620, 1989.

[Rieger, 1976] C. Rieger. On organization of knowledge for problem solving and language comprehension. *Artificial Intelligence*, 7:89–127, 1976.

[Schank and Abelson, 1977] R. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding.* Hillsdale, NJ, 1977.

[Selfridge, 1989] M. Selfridge. Toward a natural language-based causal model acquisition system. In W. Horn, editor, *Causal AI Models: Steps Toward Applications*, pages 107–128. Hemisphere Publishing Corporation, 1989.

[Simon and Hayes, 1979] H. A. Simon and J. R. Hayes. *The Understanding Process: Problem Isomorphs (1976)*, volume 1. Yale University Press, 1979.

[Winograd, 1972] T. Winograd. *Understanding Natural Language.* Academic Press, New York, 1972.

[Wong, 1981] D. Wong. Language Comprehension in a Problem Solver. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, 1981.