

13

AD-A266 135



Sponsored by Defense Advanced Projects Agency (DOD)
DARPA/CSTO

Title of Contract: "A Proposed Research Program in Strategic Computing"

Project: Shared Knowledge Base Technology (Knowledge Representation)
DARPA Order No. 6096

Issued by DSS-W under Contract MDA903-87-C-0641

Period of Performance: 09/01/87 - 08/31/92

USC / ISI

Final Rpt

DTIC
ELECTE
JUN 25 1993
S E D

CLEARED
FOR OPEN PUBLICATION

JUN 8 - 1993 3

DIRECTORATE FOR FREEDOM OF INFORMATION
AND SECURITY REVIEW (OASD-PA)
DEPARTMENT OF DEFENSE

"The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies either expressed or implied of the Defense Advanced Research Projects Agency or the U.S. Government."

93

93-13420



93-J-2092

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION USC/Information Sciences Institute	6b. OFFICE SYMBOL	7a. NAME OF MONITORING ORGANIZATION ARPA	
6c. ADDRESS (City, State, and ZIP Code) 4676 Admiralty Way Marina del Rey, CA 90292		7b. ADDRESS (City, State, and Zip Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research ARPA	8b. OFFICE SYMBOL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA-903-87-C-0641	
8c. ADDRESS (City, State, and ZIP Code) University of Southern Calif at San Diego (A-034) Scripps Institute of Oceanography 8603 LaJolla Shores Drive San Diego, CA 92093-0234		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (INCLUDE SECURITY CLASSIFICATION) KREP2			
12. PERSONAL AUTHOR(S) Robert M. MacGregor			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 92/08/31	15. PAGE COUNT 6
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	Knowledge Representation, knowledge base management system, sharable knowledge bases, concept classifier, multiple paradigm programming
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>The task of the Knowledge Representation Project was to develop a next-generation knowledge representation system for reasoning about declarative knowledge structures, together with a programming language that supports multiple programming paradigms. These features are implemented in a system called LOOM. A primary goal of the LOOM system was to develop tools and techniques that promote sharing and reuse of knowledge bases. LOOM's reasoning architecture centers around an inference engine called a concept classifier. A classifier is able to automatically organize definitional knowledge and constraint rules into hierarchies, it detects inconsistencies within a knowledge base, and it enables a richer class of deductive inferencing than that supported by traditional expert technology. The LOOM system is enabling the transition of classifier technology to a much wider class of users, where it provides a valuable service in organizing and validating knowledge-based applications. The LOOM system has also demonstrated the feasibility of combining a number of different programming paradigms within a single programming language.</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (include Area Code)	22c. OFFICE SYMBOL	

continued on back

A. Task Objectives

The Knowledge Representation Project at the University of Southern California's Information Sciences Institute (ISI) is tasked to develop a next-generation knowledge representation system for reasoning about declarative knowledge structures, together with a programming language that supports multiple programming paradigms and serves to leverage the interactions between foreign application programs and the knowledge representation system. They have combined these features in a system called Loom. The Loom system enables a "model-based" programming approach to building knowledge-based software applications: Loom is designed to facilitate construction and maintenance of explicit, detailed domain models, thus enabling a highly declarative approach to building intelligent applications. Applications constructed using Loom are easier to debug, maintain, explain, and extend than those based on current-generation technology.

A primary goal of the Loom system was to develop tools and techniques that promote sharing and reuse of knowledge. The Loom knowledge representation system includes a powerful set of tools for constructing and statically debugging both the model component of a program and certain portions of the behavioral specification. Projected benefits of this technology are that it promotes the degree to which knowledge in the form of declarative models can be shared across multiple cooperating systems, e.g., by a natural language system and an expert system; and that it increases the extent to which a model developed for one application can be reused in another application. The benefits are possible in great part because of the stress that Loom technology places on the explicit representation of knowledge.

B. Technical Problem

A typical Loom application program consists of an explicit, detailed model of the application domain, together with a procedural specification that defines the behavior of the application program. The Loom modeling language contains a rich set of primitives for constructing declarative representations of a model. This makes it possible for a relatively high percentage of a Loom application to be expressed declaratively. Loom has made a much greater investment in developing a powerful deductive facility than any of the current-generation of KR systems, and as a consequence, declarative domain knowledge plays a more significant role in Loom applications than in those built using current-generation technology.

Loom's reasoning architecture centers around an inference engine called a concept *classifier*. ISI has been at the forefront in developing classifier technology during the last decade. The classifier has proved to be a means for (1) increasing the system's deductive power; (2) enabling significant improvements in system performance; and (3) coordinating the use of multiple special-purpose reasoners within a single software architecture. The Loom language fuses four different programming paradigms, each of which exhibits model-driven characteristics: (1) in object-oriented

programming the action chosen to perform a task is the one which most closely matches the current knowledge base state; (2) in event-driven programming (e.g., a production rule system) a program can react directly to changes in the state of the knowledge base; (3) in logic programming, the logical consequences of each program decision are automatically propagated throughout the knowledge base; (4) in goal-driven programming problem solving is accomplished using heuristic search techniques.

The principal problems to be solved by the project were (i) to extend the power and scope of classifier technology to enable routine use of a classifier within intelligent applications, and (ii) to demonstrate the successful integration of multiple programming paradigms within a single knowledge representation system.

C. General Methodology

Loom is a stand-alone knowledge representation system, designed to support the execution of intelligent software applications. Loom is implemented in the Common Lisp programming language, which enables its use by other DARPA-sponsored research projects in the area of intelligent software systems. Loom technology is being validated by actual usage by other DARPA and non-DARPA projects. At the termination of this contract, Loom was being used by five other projects within ISI, and Loom had been distributed to more than 20 universities and corporations outside of ISI.

D. Technical Results

The inferential capabilities of the Loom system match or exceed those of current-generation commercial shells. Loom provides several capabilities not found in any of these systems, including (1) an ability to automatically organize and validate domain terminology; (2) an ability to reason with incomplete descriptions of domain objects; (3) an object-oriented message passing protocol that is strictly more general than those found in other systems.

Below, we list some specific results achieved during the execution of this contract.

- Developed an incremental classifier that can automatically recompile its definition hierarchy when definitions or constraints are revised. This facility permits a classifier to be applied to the task of incrementally constructing large knowledge networks .
- Extended the classifier's representation and reasoning capabilities: Added a role-relation hierarchy to complement the concept hierarchy; added support for sets, arithmetic intervals and inverse relations. Resolved a long-standing problem of how to represent necessary and

sufficient conditions by providing separate reasoning components for handling definitional and constraint knowledge. Added the ability to represent n-ary relations. Enhanced the Loom classifier with the ability to represent arbitrary disjunctive descriptions. Loom's deductive capabilities are now sufficient to support a unification-based parsing application.

- Achieved a solution to a long-standing AI problem by designing and implementing a default logic which (i) can represent a wide variety of default rules (ii) produces deterministic responses for all possible states of a knowledge base, and (iii) computes these responses rapidly enough for use within on-line application programs.
- Interfaced a reasoner for Horn rules with the Loom classifier, resulting in a hybrid reasoner for a logic that is strictly more expressive than either logics supported by current-generation shells, or logics for languages such as Prolog.
- Developed a method dispatching facility that extends traditional object-oriented programming by utilizing a pattern-driven dispatch facility, enabling method selections to be sensitive to arbitrary states in a knowledge base.
- Widened the scope of the classifier's ability to rapidly detect and circumscribe inconsistencies in a knowledge base. Loom implements a four-valued logic that enables it to perform deductions over knowledge bases containing inconsistent facts.

E. Important Findings and Conclusions

The Loom system demonstrates that a concept classifier can be successfully used in applications that operate over medium-scale domains. Before the advent of the Loom system, the use of concept classifiers was restricted to a small number of researchers. The Loom system is enabling the transition of classifier technology to a much wider class of users, where it provides a valuable service in organizing and validating knowledge-based applications.

The Loom system also demonstrates the feasibility of combining a number of different programming paradigms within a single programming language. By adopting a common declarative framework to underlie all of the implemented paradigms, the Loom system has demonstrated that multiple reasoners and programming disciplines can be made to interoperate smoothly, so that application developers who use multiple paradigms view their applications as single programs, rather as a set of distinct reasoning modules.

F. Significant Hardware Development

Not applicable.

G. Special Comments

None.

H. Implications for Further Research

The end of the current project witnessed efforts to apply Loom to applications whose requirements in terms of scale exceeded the system's capabilities. An important focus of future work will be to interface Loom to relational database management systems (DBMSs), and to object-oriented database systems (OODBS), thus enabling Loom to be used with much larger applications.

Although the Loom system has been a leader in developing classifiers that recognize increasingly expressive concept definition languages, Loom users are requesting still greater expressive power. A future version of Loom will implement a classifier able to operate on concept definitions phrased using the full first order predicate calculus.

A concept classifier is likely to prove to be a particularly valuable tool for constructing reusable domain ontologies. To be successful, however, such a tool needs to be embedded in an environment specifically tailored for this purpose. One focus of future work will be to enhance Loom with additional capabilities specific to an ontology-building environment.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Availability or Special
A-1	

DTIC QUALITY INSPECTED 3