Context: KBE (MBSE)

SPERRY　Burroughs

ARTIFICIAL INTELLIGENCE

KNOWLEDGE SYSTEM CENTER (KSC)
3001 Metro Drive
Bloomington, MN 55420
(612) 851-3369

USE inc.

routines is constructed from the set of data requirements and representative queries.

Appendix B contains a more detailed description of how we envision the designers will use the system during the DB design process.

4.4 KEE Structures Used

This section will contain information pertaining to how the various features of the KEE environment are utilized. The following is a list of the KEE features used in the system:

- Frames
- Unit
- Slots
- Methods
- ActiveValues
- ActiveImages

As was mentioned previously, the basic data structure utilized within KEE is the FRAME. This structure readily supports the network-hierarchy data representation needed to generate the LDS. The CLASS, SUB-CLASS, and UNIT concepts can be directly utilized to describe and represent the various entities, attributes and relationships. Each of these types of objects is represented in the knowledge base by a CLASS. A CLASS is created for each, with a number of MEMBER SLOTS being defined. Each of the individual entity, attribute and relationship types is represented by a SUB-CLASS and is retained in the knowledge base as a KEE UNIT. Each UNIT is represented by a FRAME.

Due to the inheritance capabilities provided by KEE, each of the individual UNITs created will inherit the necessary MEMBER SLOTS from the CLASS. This allows the user to set up UNIT classes and have all the UNITs inherit the same slots. The values associated with the inherited SLOTS will be set according to what is appropriate for the particular UNIT being defined. The slot facets can also be modified in the child UNITs by overriding the inheritance value.

In addition to inheriting MEMBER SLOTS from the CLASS, the various UNITs can also have SLOTS defined for them which are only applicable to the individual UNITs. This type of SLOT is referred to as an OWN SLOT. When a member slot is inherited down to a childUUNIT, the slot becomes an OWN SLOT and is therefore not inheritable to another level lower in the hierarchy.

Figure 6 depicts the structure of the knowledge base utilized to represent the meta-DB information. Primarily the knowledge base contains the classes ENTITIES, ATTRIBUTES, RELATIONSHIPS and VIEWS. In order to provide a graphical representation which could be further manipulated the ENTITIES.IMAGES and the RELATIONSHIPS.IMAGES classes were created.

The following are some of the member slots associated with the ENTITIES class

Special Issue on Database Design Aids, Methods and Environments, Database Engineering Review, Vol 17 No 4, Dec 1984

Sperry, Explorer System Description, publication UP-11324

Sperry, KEE, Software Development System Description, publication UP-11325

Paltin Sturdza, "Data Dictionary Design With an Artificial Intelligence Model", Proceedings of the 16th Hawaii International Conference on System Science, 1983, pp 406-412

John Switlik and Ron Short, "The Database Environment and the LISP Machine", Proceedings of AI Europa, Weisbadan, West Germany, September 1986, pp xx-xx

Tahn Joo Tan, Kan Poh Tan and Ah Moi Goh, "DATADICT: A Data Analysis and Logical Database Design Tool", Proceedings of VLDB 84, 1984

The British Computer Society Data Dictionary Systems Working Party Report, SIGMOD Record, Vol 9 No 2, Fall 1977

Toby J. Teorey and James P. Fry, Design of Database Structures, Prentice-Hall Inc., 1982

S.B. Yao, Shamkant B. Navathe and Jay-Louise Weldon, "An Integrated Approach to Logical Database Design", NYU Symposium on Database Design, May 18-19, 1978, pp 1-14

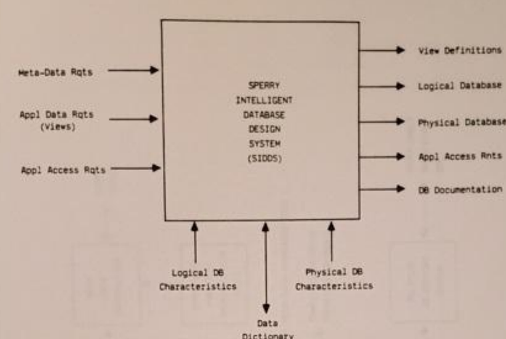


Figure 4. SIDDS

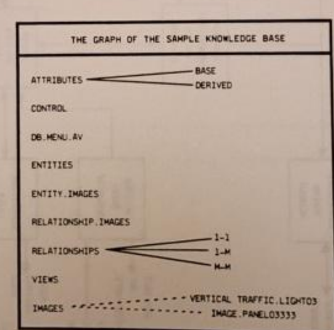
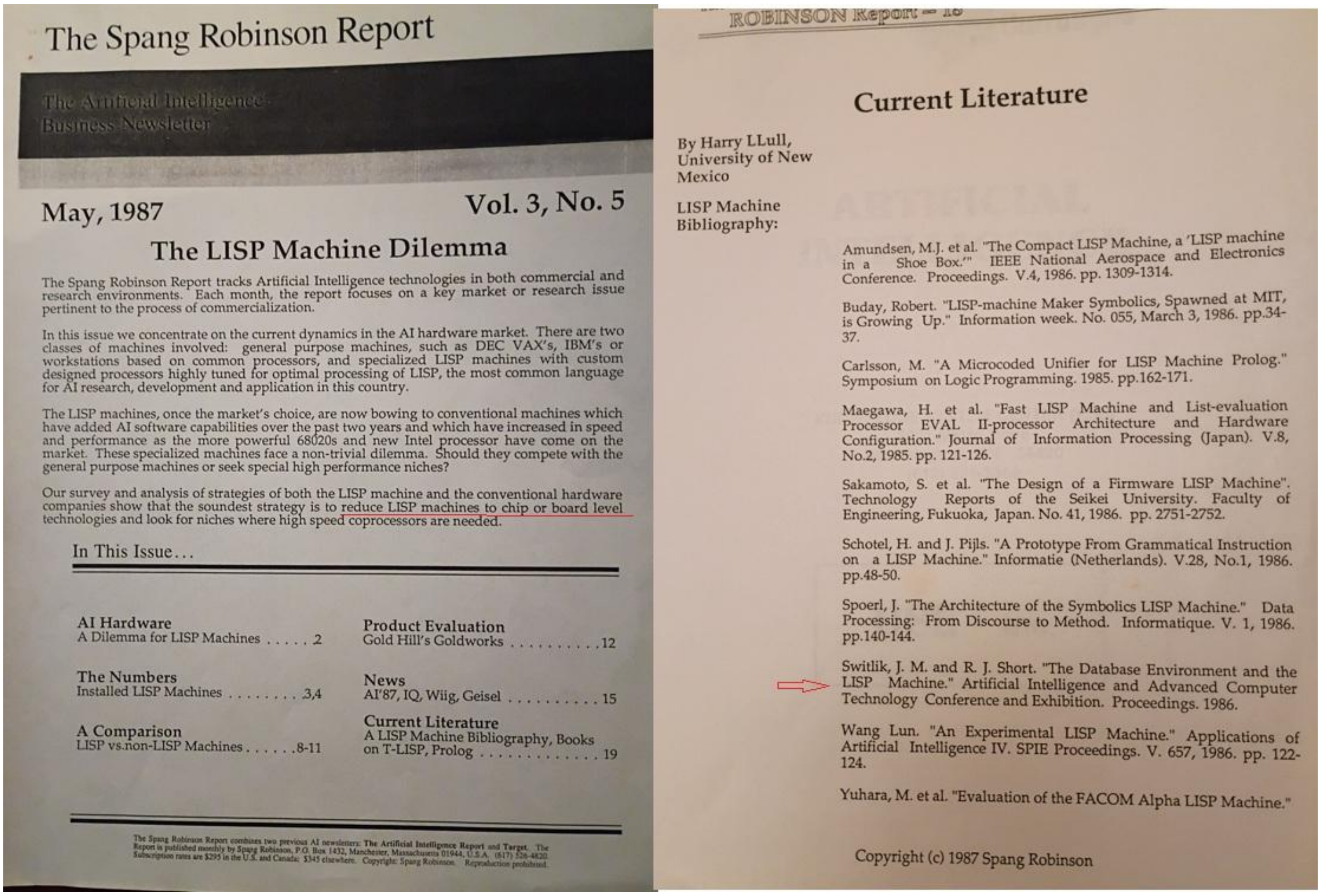


Figure 6. KEE Knowledge Base Structure for Meta-Database

KBS → UML → … → SysML

Context: KBE (MBSE)



The Spang Robinson Report

The Artificial Intelligence
Business Newsletter

May, 1987 Vol. 3, No. 5

The LISP Machine Dilemma

The Spang Robinson Report tracks Artificial Intelligence technologies in both commercial and research environments. Each month, the report focuses on a key market or research issue pertinent to the process of commercialization.

In this issue we concentrate on the current dynamics in the AI hardware market. There are two classes of machines involved: general purpose machines, such as DEC VAX's, IBM's or workstations based on common processors, and specialized LISP machines with custom designed processors highly tuned for optimal processing of LISP, the most common language for AI research, development and application in this country.

The LISP machines, once the market's choice, are now bowing to conventional machines which have added AI software capabilities over the past two years and which have increased in speed and performance as the more powerful 68020s and new Intel processor have come on the market. These specialized machines face a non-trivial dilemma. Should they compete with the general purpose machines or seek special high performance niches?

Our survey and analysis of strategies of both the LISP machine and the conventional hardware companies show that the soundest strategy is to reduce LISP machines to chip or board level technologies and look for niches where high speed coprocessors are needed.

In This Issue...

The Spang Robinson Report combines two previous AI newsletters: **The Artificial Intelligence Report** and **Target**. The Report is published monthly by Spang Robinson, P.O. Box 1432, Manchester, Massachusetts 01944, U.S.A. (617) 526-4820. Subscription rates are $295 in the U.S. and Canada; $345 elsewhere. Copyright: Spang Robinson. Reproduction prohibited.

Current Literature

By Harry LLull, University of New Mexico

LISP Machine Bibliography:

Amundsen, M.J. et al. "The Compact LISP Machine, a 'LISP machine in a Shoe Box.'" IEEE National Aerospace and Electronics Conference. Proceedings. V.4, 1986. pp. 1309-1314.

Buday, Robert. "LISP-machine Maker Symbolics, Spawned at MIT, is Growing Up." Information week. No. 055, March 3, 1986. pp.34-37.

Carlsson, M. "A Microcoded Unifier for LISP Machine Prolog." Symposium on Logic Programming. 1985. pp.162-171.

Maegawa, H. et al. "Fast LISP Machine and List-evaluation Processor EVAL II-processor Architecture and Hardware Configuration." Journal of Information Processing (Japan). V.8, No.2, 1985. pp. 121-126.

Sakamoto, S. et al. "The Design of a Firmware LISP Machine". Technology Reports of the Seikei University. Faculty of Engineering, Fukuoka, Japan. No. 41, 1986. pp. 2751-2752.

Schotel, H. and J. Pijls. "A Prototype From Grammatical Instruction on a LISP Machine." Informatie (Netherlands). V.28, No.1, 1986. pp.48-50.

Spoerl, J. "The Architecture of the Symbolics LISP Machine." Data Processing: From Discourse to Method. Informatique. V. 1, 1986. pp.140-144.

Switlik, J. M. and R. J. Short. "The Database Environment and the LISP Machine." Artificial Intelligence and Advanced Computer Technology Conference and Exhibition. Proceedings. 1986.

Wang Lun. "An Experimental LISP Machine." Applications of Artificial Intelligence IV. SPIE Proceedings. V. 657, 1986. pp. 122-124.

Yuhara, M. et al. "Evaluation of the FACOM Alpha LISP Machine."

Lisp Machine → Card (Paired with Unix)