KNOWLEDGE MANAGEMENT: A PRACTICAL AMALGAM OF KNOWLEDGE AND DATA BASE TECHNOLOGY

Charles Kellogg

System Development Corporation
Santa Monica, California 90406

ABSTRACT

This paper describes the central features of a system designed for the management of large amounts of application specific knowledge. The Knowledge Manager(KM-I) employs distinct software and hardware processors to implement:

- A file of general knowledge and an associated reasoning engine
- A file of specific knowledge and an associated searching engine

We present our reasons for believing why this can be an effective strategy for realizing many practical knowledge based/expert system applications that lie in a large overlapping area between practical AI and advanced data management technology. We then outline the major features and components of the system and discuss the range of intended applications.

I. INTRODUCTION

For several years knowledge-based system (KBS) and data management system (DMS) technologies have developed in parallel but, with few exceptions, independently of each other.

DMS systems are an outgrowth of early attempts to build practical systems for commercial and military information processing applications. The field started with the development of file and record management systems to support inventory control, general ledger, and similar applications and has progressed through the development of generalized DMS systems supporting hierarchical and network data models and, more recently, relational data models [1].

KBS systems, on the other hand, have stemmed from attempts to apply artificial intelligence research in knowledge representation and reasoning, to domains where typically small amounts of domain-specific knowledge obtained from experts in the domains, have been encoded into computer programs used to provide problem solving assistance to professionals working in the domains. Examples of seminal systems of this kind are DENDRAL, MYCIN, and PROSPECTOR. Practical KBS systems are now appearing in both commercial (e.g., R1) and military (e.g., SU/x) applications[2,3].

A number of near-term applications should be able to benefit from a merger of KBS and DMS technologies. In this paper we discuss our strategy--currently implemented at SDC in the KM-I prototype--for producing knowledge management systems that integrate capabilities of the (heretofore) separate technologies.

II. KNOWLEDGE REPRESENTATION

Most existing knowledge-based systems utilize production rules, frames, semantic nets, or semantic hierarchies for knowledge representation. These systems are usually implemented in Lisp, and all knowledge (both general and specific) is encoded into a single large list structure. Knowledge accessing and reasoning is effected by heuristic programs that work against this large list structure. Consequently, KBS applications tend to be limited by the size of LISP list structure available for encoding knowledge.

Data managenent systems in turn employ a variety of record-based data structures (nets, hierarchies, and relations) for representing specific knowledge but typically lack any form of reasoning component or means of dealing with general information. They are, further, usually programmed and optimized in conventional programming languages and microcode to make efficient use of sophisticated dictionaries and directories pointing into very large volumes of information systematically arranged on low-cost storage media.

Our approach to realizing a knowledge management system is based on:

- A solid theoretical foundation that supports the underlying knowledge representation, the means for reasoning with it, and a useful distinction between general and specific knowledge.
- State-of-the art database machines and Lisp machines that allow both powerful and cost-effective systems.

- A performance oriented reasoning engine that can derive plans from a general knowledge base to intelligently control database machine search of a specific knowledge base.

The theoretical foundation is the first order predicate calculus (with functions)-a formalism that has been both praised and condemned throughout the history of AI. Although we cannot convince critics in this short paper, some idea of the descriptive adequacy and practical utility of this calculus for knowledge and data management can be found in recent papers on "Logic and Data Bases"[4,5] and "Logic Programming"[6].

Predicate calculus theory includes both a theory of reasoning(proof theory) and a theory of semantics(model theory). The former is the basis for our reasoning engine and the latter is practically realizable through database structuring used in relational data models (and database machines that implement them).

In KM-I, general knowledge is represented as a set of logically formattted assertions(premises or logical rules) that constitute a "micro" theory about some aspect of the world, while specific knowledge, formulated as relation tuples, provides a model or interpretation of the "micro" theory.

The distinction between general and specific knowledge is sometimes made using the terms "intensional" and "extensional" after Carnap [7]. Thus the phrases "intensional database" for formalized general knowledge and "extensional database" for specific knowledge are sometimes employed.

The conceptual adequacy of our knowledge representation thus derives from logical theory but the practical utility of our approach must be judged on the basis of empirical performance of its components, separately and in combination, as discussed in Section III.

III. THE KM-I SYSTEM

Current and future configurations of KM-I are illustrated in Figures 1 and 2.

In the current version of KM-I the reasoning engine and general assertions are implemented within Interlisp and operate on a Xerox 1100 Scientific Information Processor (Lisp Machine). The searching engine is a relational data base machine that uses specialized hardware to cost effectively implement the relational model of data--the Britton Lee IDM(Intelligent Database Machine)-500[8]. Both reasoning and searching engines are designed to be general purpose and application independent within the constraints of their underlying knowledge representations.

At present communication between these two specialized machines must be mediated by a VAX 11/780. The 1100 communicates with the VAX by way of an Ethernet connection, while the IDM communicates (temporarily) by way of a serial, direct line to the VAX. The two VAX programs shown in Figure 1 convert data access strategies formulated by the reasoning engine in IDL(Intelligent Data Language) into IDM commands and, for the other direction, convert data retrieved by the IDM into Lisp Machine format.

Explicit specific assertions(facts or tuples) contained in an extensional data base are stored on a pair of 80MB disks that are part of our IDM installation, while general assertions(premises or logical rules) are stored within the 23 MB disk that is part of the Xerox 1100. (This is sufficient storage for thousands of rules and millions of tuples.)

User queries are submitted in an English-like form to KM-I's reasoning engine which translates the queries into the required logical formalism. The file of general knowledge is then employed to construct an "inference plan" (skeletal derivation). Then KM-I produces a "search/compute plan" (access and procedure-activation strategy) to be used by the searching engine for locating and manipulating relevant specific knowledge. Found information is fed back from the searching engine to the reasoning engine which uses it to instantiate the skeletal derivation and to produce an answer and an "evidence chain" (the chain of reasoning steps that leads to the answer).

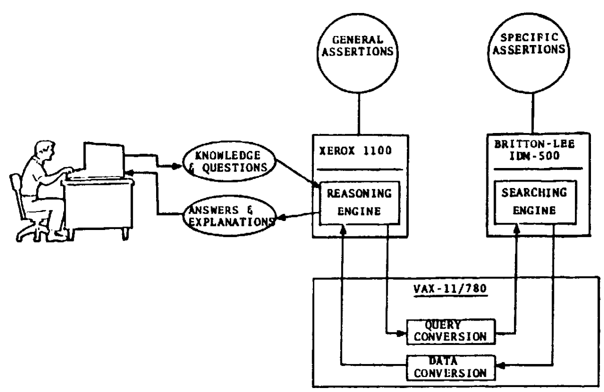

Figure 1. KM-I Current Configuration

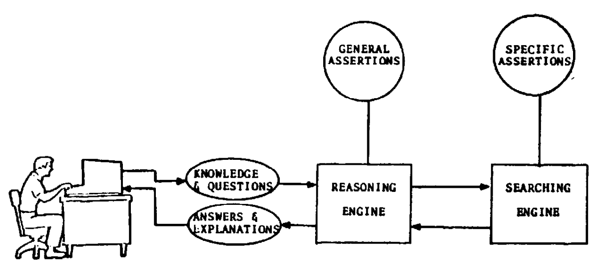

Figure 2. KM-I Future Configuration

The key words "find", "what-if", "given --- find ---" are used to trigger different forms of reasoning. "Find" queries produce backward reasoning steps that result in skeletal derivations and data base values that support desired conclusions (goals), while "what-if" queries cause forward reasoning from assumptions through the general rules and the database of specific facts.

A "given--- find---" query such as the following example from an implemented shipping/receiving application:

Given Emperor ships heavy freight to Ambassador
and Excelsior ships heavy freight to Excello
find that Emperor and Excelsior are competitors
in Valley Acres

results in reasoning from assumptions toward goals and vice versa(sometimes called "middle-term reasoning" see[9]). In the above example, the conclusion that Emperor and Excelsior compete with each other depends on the assumptions as well as on stored knowledge, both general and specific.

IV. THE KM-I REASONING ENGINE

Over the past few years we have implemented a series of deductive processors designed specifically to support question answering(see refs. [9] through [16]). The reasoning engine employed in KM-I is the latest version of our DADM(Deductively Augmented Data Management) deductive processor.

The DADM deductive processor has been designed to use relational data base systems for the storing and retrieving of specific knowledge. To support knowledge management, the design is being augmented with new acquisition and verification capabilities to more fully support knowledge base administrators. It will, for example, provide for general-knowledge extensions to existing relational databases through their schemas as well as support the acquisition of combined general and specific knowledge bases from their inception.

DADM has been used to implement and experiment with knowledge manipulation in a number of application areas including bibliographic retrieval, alternative route finding, and command and control. Its deductive pathfinding and planning strategies have proved effective in quickly locating premises relevant to user requests, in constructing required derivations, and in producing intelligent data access strategies. The first KM-1 application will be a "Manager's Assistant" which combines expert planning knowledge and a corporate data base for the assistance of corporate project monitoring and planning.

V. NEXT STEPS

We expect to bring up several other applications in the current KM-I environment but at the same time we are reviewing other Lisp and Database Machine combinations. Eventually we plan to move to the configuration illustrated in Figure 2. This will provide a "standalone" knowledge management capability not dependent upon a large mainframe computer. While such a configuration is not now inexpensive, within the next few years such a combined system could be achievable at costs comparable to those of current professional workstations.

6. REFERENCES

[1] Codd, E. F., A Relational Model of Data for Large Shared Data Banks, Comm. ACM, Vol. 13, No. 6, June 1970.

[2] Barr, A., Feigenbaum, E., The Handbook of Artificial Intelligence, HeurisTech Press, 1981.

[3] Webster, R., Miner, L., Expert Systems: Programming Problem-Solving, in Technology, Vol. 2, Jan/Feb 1982.

[4] Gallaire, H., Minker, J., eds., Logic and Data Bases, Plenum Press, 1978.

[5] Gallaire, H., Minker, J., Nicholas, J., eds., Advances in Data Base Theory, Volume 1, Plenum Press, 1981.

[6] Kowalski, R. A., Logic for Problem Solving, North Holland Pub. Co., 1979.

[7] Carnap, R., Meaning and Necessity, Univ. of Chicago Press, 1956.

[8] IDM 500 Software Reference Manual, Version 1.3, Britton Lee, Inc., Los Gatos, Ca., Sep. 1981.

[9] Kellogg, C., Travis, L., Reasoning with Data in a Deductively Augmented Data Base System, in Advances in Data Base Theory, Volume I, Gallaire, H., J. Minker, eds., Plenum Press, 1981.

[10] Kellogg, C., Research on Deductive Inference for Large Data Bases, System Development Corporation, TM-6263, 1979.

[11] Kellogg, C., Klahr, P., Travis, L., Deductive Planning and Pathfinding for Relational Data Bases, in Logic and Data Bases, Gallaire, H., Minker, J., eds., Plenum, 1978.

[12] Kellogg, C., Klahr, P., Travis, L., Deductive Methods for Large Data Bases, Proc. Fifth International Joint Conference on Artificial Intelligence, MIT, 1977.

[13] Kellogg, C., Creating Inference Plans for Deductive Access to Relational Data Bases, in Proceedings of Workshop on Logic and Data Bases, C.E.R.T., Toulouse, France, 1979.

[14] Kellogg, C., Klahr, P., Travis, L., A Deductive Capability for Data Management, in Systems for Large Data Bases, Lockemann, P.C., Neuhold E.J., (eds.), North-Holland Publishing Co., 1976.

[15] Klahr, P., Kellogg, C., Travis, L., A Deductive System for Natural-Language Question-Answering, in Natural Language Based Computer Systems, Bole, L., Ed., Hanser-Verlag, 1980.

[16] Klahr, P., Planning Techniques for Rule Selection in Deductive Question-Answering, in Pattern-Directed Inference Systems, Waterman D. and P. Hayes-Roth, Eds., Academic Press, 1978.