Large Knowledge Based Systems: An Efficient Approach

Gerard T. Capraro  Calvin Cheung  Raymond A. Liuzzi
Kaman Sciences Corporation  Kaman Sciences Corporation  Rome Air Development Center
258 Genesee Street  2800 28th Street  Griffiss AFB, NY 13441-5700
Utica, NY 13502  Santa Monica, CA 90405

ABSTRACT
The contents of this paper describes an architectural
approach that interfaces very large heterogeneous
databases with a knowledge based system. The
databases contain the extensional portion of the
knowledge base. The interface of the two systems is
accomplished by utilizing information about the
procedures contained within the knowledge based
system. The developed interface system is based on a
Prolog built knowledge base.

This research was supported by the Rome Air
Development Center under contract F30602-85-0228.

1.0 INTRODUCTION
This paper contains a description of an efficient
approach for building a large knowledge based
system. Efficiency is considered in the building of the
knowledge base and in its operation. It is assumed
that (a) the extensional or fact base comprises the
major portion of a large knowledge base, (b) these
facts can be obtained more efficiently with Database
Management Systems (DBMS) and their associated
tools, rather than an AI workstation, and/or (c)
databases already exist within the organization that
contain the majority of the required fact base.
Therefore, it is important to consider efficient ways to
interface databases and DBMS's with a knowledge
based system.

Once the knowledge based system and the databases
are integrated, an efficient method is required for
their operation. To achieve an efficient operational
architecture, it was determined that information
about how the knowledge based system utilized its
knowledge base was needed. To obtain this
information, a knowledge base system, or building
tool, would be required. It was decided that since
Prolog's computational ability and general
understanding is known by the community, that it
would be a good choice for this study. By studying
Prolog and its theorem proving methods, the needed
information was obtained to build an efficient
knowledge and database operation.

The next section contains a review of the literature.
This review provides the basis for motivating the
reader of why this work should be performed. The
contents of the following section deals with describing
the approach of how we can efficiently integrate
knowledge bases and heterogeneous databases. Since
the approach has not been fully implemented, the
next portion of the paper covers our current thinking
of how to implement the approach. The last major
section of the paper provides our conclusions, future
directions, and a summary of our work.

2.0 MOTIVATION
Knowledge based systems built today with
conventional computers or Artificial Intelligence (AI)
workstations, e.g., Lisp machines, are challenged by
the need to store and process large amounts of
intensional data, i.e., rules and theorems, as well as
extensional data, i.e., facts. Once primary storage is
exceeded and secondary storage is necessary,
processing performance is significantly degraded.
This additional memory, in many cases, comes from the
need for large data resources that require
maintenance by conventional database management
systems. Therefore, some operational knowledge
based systems need to efficiently interact with existing
DBMSs. According to Fox and McDermott[1]:

The issue of how to manage large volumes
of data has been largely ignored by AI. The
approach traditionally taken is to increase
the address space and store all the data in
virtual memory, letting the operating
system worry about the rest. With the
application of AI to problems such as
factory management, the need to store and
access large volumes of data outside of
virtual memory is critical.

The implication is that Knowledge Base System (KBS)
technology applications require database (DB)
solutions. Some of the data requirement issues
associated with these solutions in the context of KBS
technology are outlined below.
Inheritance and the more general, relation-based, search also raise data management issues. The cost of such searches increases dramatically when much of the knowledge is not contained in main memory but in a DB. If such operations are to be supported, then we must reduce the time for swapping data, so that it appears invisible. Better database fetching must also be sought. Database systems available today do not appear to provide a solution; access times to records are quite high, relative to memory access times. Solutions that include parallelism, multiple levels of caching, and intelligent pre-fetch need to be investigated.

In concert with the above authors, [2] addresses the database aspects related to knowledge based systems. He states that:

In the problem solving processes described above, not only the intentional data, but the extensional data as well, should be used effectively. In a logical sense there is no substantial difference between them, because each represents relationships among entities. From an operational point of view, however, there are important distinctions; because of these distinctions, there are good reasons to separate them physically in a system. First, the intentional data is available for deductive operation because of its applicability and generality in representing relationships, while the extensional data includes only the instances usually used in program execution at the final state of processing. Hence, their accessing modes are different. Second, since the extensional data set tends to grow very large, an efficient storage method is necessary. Finally, a number of databases have already been created. It is desirable that these databases be used as they are.

Note also, the concern over the use of very large knowledge bases that was expressed by J.A. Robinson [3] during an ICOT Public Lecture given in Tokyo. He in essence said that the systems that have been built in logic programming have been small, in that they execute within one machine. He contended that in order to reach speeds of 109 Logical Inferences Per Second (LIPS) by 1990, all the processing and data would have to be done within fast memory. If the Fifth Generation Project expects to develop architectures capable of performing at gigabytes of speed while operating on large database collections (e.g., in the terabyte range) on secondary storage, then significant problems will have to be overcome. Robinson's comments only address the speed and memory issues. He didn't address how one maintains the dynamic aspects associated with managing a terabyte-sized database. The DBMS community has been building tools for years to address some of the problems that may eventually be employed by the knowledge base community in its attempt to construct very large knowledge bases.

The intensional portion of the knowledge base, once built and accepted, does not change as frequently as the extensional portion. The extensional portion of the knowledge base may have its contents added to, deleted, and/or have the value of its facts changed. These updates to the extensional portion of the knowledge base may occur seldomly or, in some cases, may happen very frequently and randomly, similar to, for example, a database in an airline reservation system. Another aspect that can stress a knowledge based system possessing a very large extensional and dynamic database is the process of obtaining this data. If the data exists within a single database, then the interfacing of the two system components to form an efficient integrated system, is challenging. However, if the data required exists within more than one database maintained by different database management systems, the approach to integrating these systems is complicated and more challenging.

2.1 Synopsis

A compilation of the major concepts and suggestions made by the above authors is synthesized:

1. Intensional and extensional data should be physically separated,
2. Extensional data tends to increase,
3. It is desirable to utilize existing databases (and database management systems),
4. Current database systems by themselves are not sufficient for performing searches for data in a knowledge based system environment,
5. Time for swapping data from a database to a knowledge base must appear invisible to the knowledge base user, and
6. Data search techniques that include parallelism, multiple levels of caching, and intelligent pre-fetch should be investigated.

The implementation and integration of these suggestions and concepts are necessary if very large extensional knowledge bases are ever to be efficiently built and used effectively. If a knowledge based system with a very large extensional knowledge base is built from scratch, or portions obtained from data bases within the organization, then the integrating of data and knowledge based systems must be accomplished in an efficient manner. The objective of this paper is to present an approach of integrating these two systems. The approach addresses all of the above suggestions in an architecture, that will provide fast responding very large knowledge bases.
3.0 APPROACH

Our basic premise is either (a) the developer of a knowledge based system has the majority of the factual portion of the knowledge base digitized within his/her organization, or (b) can build this factual base easier with a data base management system and its associated hardware and software tools, than using an AI workstation. Once the factual data portion of the knowledge base is within a database system, then accessing the extensional portion of the knowledge base by the knowledge based system in a timely manner is required. A solution of the first part of the problem is to develop an integrated database system for all the databases within the organization. The solution of the second part of the problem is based on knowing the knowledge based system’s needs and processing procedures for the extensional portion of the knowledge base. This is necessary in building an efficient architecture that accesses the extensional data fast enough so that it will appear to the KBS user as if the total knowledge base is resident within the knowledge base workstation/computer.

A solution for developing a fast responding architecture for a very large knowledge based system is based upon integrating database and knowledge base technology. Since we are assuming that the extensional portion of the knowledge base is very large, we are concerned about integrating DBMSs. We are also concerned with speed in accessing these data from the databases and hence are interested in knowing how the knowledge based system processes its knowledge base. The first portion of this section describes a method of integrating DBMSs that can act as a “backend” to provide the factual portion of the knowledge base. A similar method is applied to integrate the extensional portion of the knowledge base with the integrated databases. This provides a logical view of how to interface the databases with the factual portion of the knowledge base. The next portion of this section provides a description of the knowledge required to develop an architecture that can provide the factual portion of the knowledge base to the KBS. The last portion of this section describes the integration of the KBS and the integrated DBMSs.

3.1 How to Integrate the Data

An architecture for multiple databases acting as a backend to a knowledge base requires data from different data/knowledge domains, see Figure 1. This figure represents N separate DBMSs being used separately within the organization. Assume everything below the dotted line is currently in operation within the organization. A proposed solution to the problem would be to create a Global Data Dictionary/Directory. This Global DD/D could be part of a Global DBMS which would provide a total view of the data assets to interested participants in the organization. This approach is an example of what is sometimes called an integrated database approach. Efforts studying the integration of heterogeneous databases are being funded by the Navy, Defense Advanced Research Project Agency (DARPA), USAF, and the National Bureau of Standards (NBS). See [4, 5, 6, 7] for detailed information on current ongoing projects for integrated systems.

Since these databases are non-homogeneous it can be assumed that they most likely reside on different computers, different DBMSs, and have different representations at all levels, i.e., machine, logical and physical. That is why the mapping functions $A_1, A_2, \ldots, A_N$ are shown in Figure 1. These mapping functions allow the Global DD/D to communicate to the different DBMSs. These mapping functions must be able to communicate with each DBMS using its I/O protocols, syntax, and semantics such that they appear to each DBMS as ordinary users.

The Global DD/D must contain a standard structure for all the attributes in all the databases. It must also have the ability to convert its schema representation to all the individual schemas associated with each database. The Global DD/D may be required to also be responsible for example, data integrity, security, synonyms and homonyms between all the databases. Its most difficult problem, however, will be to maintain the relationships of all the schema structures in the different databases and to communicate via the “A” mapping functions.

3.2 KBS and Integrated Databases

If a backend computer, integrated database system, could be built, an AI workstation would interface with the Global DD/D via its own connection function “C” [see Figure 1]. This function, similar to the $A$, mapping functions, could map the different data values or facts to the knowledge base and the Global DD/D could provide the mapping function from the GDBMS’s schema to the “schema” required by the knowledge based system. e.g., frame representation, logic, rules,
and semantic nets. This mapping function could be exercised periodically updating the knowledge base, or it could be done on demand when the system is not in use. This integration capitalizes on existing databases and maintenance functions that are being done by the users of the current established DB system. The knowledge base is taking advantage of the system by moving all of the data update, data gathering, and data integration functions to the DB area that exists within the organization. These mapping functions along with the Global DD/D provides a logical view of how the extensional portion of the knowledge base can be integrated with the databases.

3.3 What About Speed?

The interest of this paper is to develop a feasible design for the function (C) of one knowledge based system with an integrated data based system. The design assumes that the intensional portion of the knowledge base is totally resident in the workstation and the extensional portion of the knowledge base is partially or totally resident within the integrated database system. The logical structure defined so far, if implemented in software, would provide facts by formulating simple queries from the AI workstation to the database system. This would most likely result in too slow of a response for the knowledge base system. Hence, if pre-knowledge of the facts required to satisfy a requested goal is known, then an algorithm and architecture can be developed that will provide the extensional data fast enough such that the total knowledge base will appear to reside within the workstation.

3.4 Why Prolog?

To determine the feasibility of this approach, an in-depth understanding of the knowledge based system's processing procedure must be known. The procedure chosen was Prolog, because of its known computational ability and general understanding by the community. However, it is believed that a similar approach can be developed for any knowledge based system whose processing procedures are well understood. To appreciate the algorithm and architecture system, it is important to understand the important features of Prolog.

3.5 What Intelligent Data Are Needed?

Figure 2 illustrates a general structure for a very large knowledge base. In this figure the extensional portion of the knowledge base is shown around the circumference of the figure, and the intensional portion shown in the center as circles and "flattened" cylinders. The largest constituent of the knowledge base is assumed to be the extensional (factual) data. The intensional data is assumed to require a small portion of the total memory requirements. The purpose of this illustration is to convey to the reader the important information that is needed in order to separate portions of the knowledge base when the knowledge base is too large to be contained within one machine's main memory.

This needed information includes the names of the predicates within each of the circles and cylinders, their relationships and the names of all the facts, and their relationships. The arrows between the circles indicate the relationship between the head of a clause and the predicates in its body. The arrows between a circle and a rectangle indicate that the predicate has one or more facts in the extensional portion of the knowledge base. In general, these arrows can be thought of as the procedure calls in the Prolog theorem prover. However, because facts do not have any predicates in their bodies, the only way they can be accessed is from a procedure call from a predicate within a circle, or from a goal submitted by the programmer.

This information is tentatively considered as necessary to build an algorithm that could provide a fast search capability on the extensional portion of the knowledge base contained on secondary storage. When a query or goal is submitted to the knowledge based system, the same goal would be sent to a special architecture that would have the needed data, (as shown in Figure 2), resident in software and hardware. The special architecture would find the goal the programmer requested and process its data. One of the processes it would perform is to partition all of the extensional knowledge base into two portions. These portions would be, (1) all those facts that are connected to the goal that was submitted, and (2) those that are not related in any way with the submitted goal. Portion (1) would then be retrieved from secondary storage and transferred to the AI workstation.
Consider, as an example, that a goal submitted by a programmer is the shaded circle shown in Figure 3. The special architecture would obtain the requested goal from the programmer and then partition the extensional database by dividing the total knowledge base into portions A and B as shown. Then, in so doing, the architecture "knows" that the portion of the extensional knowledge base that may be necessary for the theorem prover to satisfy its goal is contained in the facts shown as F1, F2, and F3. These facts can then be accessed from secondary storage and made available to the main memory in the knowledge based system for the theorem prover to access and solve the submitted goal.

4.0 IMPLEMENTATION

The knowledge required to interface the knowledge base system with the integrated data base as shown in Figure 1, was presented by representing the Prolog intensional and extensional knowledge base as one large complicated graph. This data represented by the graph would have to be obtained from the description of the knowledge base in the AI workstation. In addition, the extensional portion of the knowledge base would have to be designed to interface properly with the integrated data bases. The implementation requires that two algorithms be built, one that can generate this large graph and one that operates upon the graph by finding the facts that are needed. The present implementation design is composed of an Intelligent Knowledge Base (IKB) Retriever, a secondary storage device loaded with the factual portion of the knowledge base, and a software interface to the integrated data base.

4.1 Intelligent Knowledge Base Retriever

The approach for solving the large knowledge base access problem is shown in the top portion of Figure 4. The facts of the knowledge base are resident on secondary storage and are controlled by two sources, the Intelligent Knowledge Base (IKB) Retriever processor and the KBS. The IKB Retriever could be resident on a conventional computer or workstation, or developed as a special architecture. The bottom portion of Figure 4 represents the connection to the integrated architecture shown in Figure 1 and the top portion represents a more detailed definition of the connection function C shown in Figure 1. Two major changes have been made in this figure and will be discussed in the next section.
remaining portion of the fact base to the KBS. All of these approaches can be implemented. Models have been developed to assess which one is most appropriate given the dimensions of the problem or knowledgebase specifications.

4.2 Connection to the Integrated Database

The lower portion of Figure 4 shows two major changes to the previous architecture illustrated in Figure 1. One is the addition of a Knowledge/Database Manager, who is responsible for the Knowledge and Databases in the organization. His/her management functions would be similar to those allocated to a Database Manager or Administrator, except for the additional responsibility of managing the knowledge base. The other change is to the Global Data/Dictionary/Directory. The GDD/DD will have the additional data representing the factual portion of the knowledge base and hence, is now referred to as the Global Data/Dictionary/Directory (GDD/DD). In addition, it will require a mapping function (M) necessary to map the updated databases to secondary storage media for the factual portion of the database. Its purpose is to map updates to the knowledge base as they are requested by the K/D Base Manager or through the KBS resident computer.

The additional changes to the GDD/DD in order to add the factual data representing the format, conversion of data fields, relationships to the other databases in the heterogeneous databases, etc., appears to be as easy as any other database that is added to the integration. The factual portion of the knowledge base will be characterized as if it were a database. In some respects, it appears to resemble a relational schema.

5.0 CONCLUSIONS, FUTURE DIRECTIONS, AND SUMMARY

The knowledge/data integration approach presented addresses the major issues, concepts, and suggestions brought forth by researchers involved in the analysis and development of knowledge bases. These considerations, expressed above, and our responses, are listed below.

"The intensional and extensional data should be physically separated." While this design guideline is generally followed, a small portion of the intensional data is retained with the extensional data to provide information needed by the Intelligent Knowledge Base Retriever to perform efficient fact retrieval. These operations are intended to speed KBS processing on a large fact base resident on secondary storage.

"Extensional data tends to increase. It is desirable to use existing databases." The architecture presented in this paper aims at accommodating the increases in extensional data growth by employing integrated database management. In applications involving a single DBMS, this approach can also be employed.

"Current database systems by themselves are not sufficient for performing searches for data in a knowledge based system environment." The architecture is presented in light of the generality of this statement. It features intelligent search capability which can be implemented in both hardware and software to meet specified performance requirements.

"Time for swapping data from a database to a knowledge base must appear invisible to the knowledge base user. Data search techniques that include parallelism, multiple levels of caching, and intelligent pre-fetch should be investigated." The factual portion of the knowledge base under the architecture is not managed by the DBMS when the knowledge base is being accessed. It, however, does provide for the maintenance of the fact base and periodically updates the fact base as determined by the Knowledge/Database Manager. The architecture features parallelism in that the Intelligent Knowledge Base Retriever is operating in parallel with KBS. As the Retriever design progresses, it too may require parallelism. The architecture is designed to provide intelligent pre-fetch capability by utilizing a subset of the intensional portion of the knowledge base. One or more levels of caching between the fact base and the KBS memory may be needed. One level of caching, requiring that some of the instances of all of the facts be kept in the KBS, was presented in the discussion.

5.1 Present Status and Directions

To determine if the architecture approach presented is truly a feasible one, more information is required concerning the development of the IKB Retriever and its interface with its secondary storage and the AI workstation. A modeling study is completed of the Prolog graph representation of its knowledge base, as depicted in Figures 2 and 3. The model is based on Graph Theory where the knowledge base graph is described as a Node-Arc Incidence Matrix. With this representation, an algorithm has been formulated that efficiently determines all of the extreme nodes (facts) in the Directed Linear Graph that are connected "after" a given node in the graph. This algorithm is the main function of the IKB Retriever for determining the facts that are required given a goal from the user of the KBS.

In addition to developing this algorithm, effort has been expended in modeling the data storage and retrieval procedures for interfacing the IKB Retriever and the secondary storage devices. Timing equations are being finalized to determine the best methods for interfacing the AI workstation, IKB Retriever, and the storage of the facts on the secondary storage devices.

5.2 Summary

The contents of this paper proposed an architecture for an integrated data/knowledge based system centered on a global data/knowledge dictionary and directory. The process of developing a KBS based on Prolog implementation schemes was studied and an architecture proposed which allowed the extensional (fact) portion of the knowledge base to exceed the
KBS resident system's memory. Under this architecture approach, the extensional portion of the knowledge base was integrated with the databases into a combined data/knowledge based system.

ACKNOWLEDGEMENT

This work was sponsored by the Rome Air Development Center, Griffiss AFB, New York, under contract F30602-85-0228. The authors would like to thank Mr. Allen V. Lazzara, James M. Marcinkowski, and Mr. John M. Decker for their contributions.

REFERENCES