Conceptual Information Processing -- A Robust Approach to KBS-DBMS Integration*

Allen V. Lazzara and William Tepfenhart, Ph.D.
Knowledge Systems Concepts, Inc.
262 Liberty Plaza
Rome, NY 13440-4460

Richard C. White
Datalogic Systems, Inc.
2790 Skypark Drive, Suite 110
Torrance, CA 90505

Raymond Liuzzi, Ph.D.
Rome Air Development Center/COTD
Griffiss AFB, NY 13441-5700

ABSTRACT

Integrating the respective functionality and architectural features of knowledge base and database management systems is a topic of considerable interest. This paper addresses several aspects of that topic and the associated issues, beginning with a discussion of the significance of integration and the complexity of the problems associated with accomplishing such integration. Current approaches to integration are described as well as their shortcomings. These shortcomings and the need to fuse the capabilities of both knowledge base and database management systems motivates the investigation of information processing systems based on new information processing paradigms. One such paradigm is concept-based processing, i.e., processing based on concepts and conceptual relations. Reporting on ongoing work, this paper describes our approach to robust knowledge and database system integration by discussing our progress in the development of an experimental model for conceptual information processing.

1 Introduction

Knowledge based systems (KBS) and database management systems (DBMS) technologies have evolved essentially independently of each other. Until very recently, there has been little concern for the mutual integration of these technologies and their respective systems. But with the advent and evolution of numerous experimental knowledge based systems in industry, government, and the military, most of which were developed to establish a proof of concept for automating the performance (or portions thereof) of specific user functions independently of any operational environment, developers have recognized the potential for enhancing current information systems with advancing knowledge based system (KBS) techniques and methods. Since most information systems provide functional support to users through large, shared, database systems, the insertion of knowledge based systems or the integration of knowledge based system techniques into these mainstream environments has resulted in considerable complexities. Despite the current attention to that problem as evidenced by the numerous articles, projects, and discussions devoted to it, there is a conspicuous absence of any definitive successful solutions.

Our examination of recent KBS-DBMS integration efforts suggests that fundamentally different approaches to the fusion of the respective DBMS and KBS capabilities must take place. Although it is clear that the total convergence of KBS and DBMS technologies is many years away, our research has

----

* This research is supported by Rome Air Development Center under contract F30602-85-C-0228.

been directed toward the development of concept-based information processing that results in systems that take on the characteristics of both knowledge and data systems. Our approach to the development of such systems is from an infological, as well as a computational perspective. With regard to the infological, we are attempting to define methods for applying and verifying the concepts and conceptual relations in our system. From a computational perspective, we have been building a library of conceptual and computational primitives and strategies to realize concept based processing. The theoretical basis of which has been formally established in Sowa's conceptual graph formalism.

The paper begins with a synopsis of KBS-DBMS integration R&D trends and an overview of the integration problem. Current integration implementation alternatives are described. Reporting on work in progress, it concludes by discussing our initial steps towards developing a conceptual information processor, and its infological basis.

2 Trends in KBS-DBMS Integration

Researchers in artificial intelligence, database management, software engineering, and machine architecture, as well as numerous other disciplines, have contributed theories, methods, and questions concerning the development of advanced information systems. A principal concern in many of these studies is the linkage of knowledge based and database system technologies. Concentrating on the KBS and DBMS technologies, a diversity of views on KBS-DBMS integration exist. For example, researchers such as Fox and McDermott have addressed the operational issues of inserting knowledge based systems into the heterogeneous and data rich computational environments of manufacturing [Fox 86]. Jarke and Vassilou have reported on a variety of KBS-DBMS interconnection possibilities [Jarke 85]. KBS-DBMS application techniques such as the extraction of knowledge from existing clinical databases have been attempted by Wiederhold and Blum [Wiederhold 86], while Missikoff and Wiederhold have defined criteria and requirements for an expert database system [Missikoff 86]. Bic and Gilbert have taken the view that more expressive data modeling formalisms are necessary to enhance database management systems, see [Bic 86]. Kellogg and many others, including the Japanese, continue their research in logic-based systems, as a vehicle to progress from data management to knowledge management, see [Kellogg 86].

There is some evidence to indicate that DBMS technology is being influenced by KBS technology. Research in DBMS data models is shifting from the development of traditional machine-oriented database modeling formalisms to more expressive models of the user problem domain. Efforts are being made to distinguish data from conceptual models applied to that domain. Thus, within DBMS technology there is the recognition that the work by AI researchers in conceptual modeling, knowledge representation, inference processing, natural language processing, and graphics-oriented user interfaces is relevant for expanding DBMS capabilities. Hence, DBMS researchers are attempting to evolve operational environments which support different user views on the content and organization of their data. In addition, there is considerable interest in producing "intelligent" database systems with capabilities for automatic data acquisition, processing, and dissemination without user intervention.

KBS technology can be effectively applied today if the problem to be solved is narrowly-scoped and its solution by an expert can be captured. Although no common definition or criteria exist for knowledge based systems, [Harmon 86] delineates knowledge based systems that have been developed for business and industrial contexts. Despite the fact that these systems are touted as 'operational', the majority are for very special and limited applications that are not well integrated into the context of larger information systems. It has become apparent that future KBSs cannot exist as such. That is, they must exist within the context of much larger systems with significant interfaces to their subsystems, e.g., database management and communications subsystems. A similar conclusion applies to the KBS development tools. In most instances, these capabilities are extremely "low-level"; that is, skilled programming expertise is necessary to build application solutions from the primitive capabilities provided by the tool set. The languages and tool sets used to develop these systems, e.g., ART, KEE, OPS5, exhibit weaknesses reflecting their state of development and motivation to solve small-scale, narrowly-scoped problems. In addition, they do not support application scaling, meet certain kinds of system performance requirements, and integrate with conventional computational and database processing. Work is ongoing to extend the functionality of these existing tools, e.g., the extension of Intellicorp's KEE to OPUS, and develop next generation tool sets, e.g., Teknowledge's ABE. These extensions are in response to the near-term need to increase the size and performance of knowledge based systems, integrate with conventional database and software systems, modularize and reuse components, and share knowledge bases among several related applications.

Another significant trend overlapping both KBS and DBMS technologies is the movement of KBS development tools, i.e., expert shells, to mainframe environments. The motivation is to place KBS technology in the environment of large scale general purpose computers, thus, increasing the potential for integration with mainstream software and databases. To this end, IBM has introduced the Expert Systems Environment (ESE), an expert systems shell that runs on IBM mainframes under MVS and VM operating systems. Also, Software A&E has ported its Knowledge Engineering System (KES) expert system shell to IBM mainframes, and other vendors are following. While it is too early to assess the repercussions of this movement, it is likely that they could in the near term have a profound influence on the integration of KBS capabilities into those environments.

3 The Complexity of the Integration Problem and Approaches

As discussed in [Lazzara 86], the complexity of the KBS-DBMS integration problem stems from several fundamental differences in the two kinds of systems, including the type of problem each is intended to solve, their period of development, and basic design/implementation features. In addressing some of these differences, the subsequent discussion will focus on the reasons for that problem's complexity, as well as some of the objectives to be achieved in developing a solution to it. Since our experience is in the study and development of prototype Command Control, Communications, and Intelligence (C^3I) knowledge and data systems for the United States Air Force, the remainder of this paper will address this operational domain.

At this point, the discussion will assume a naive interpretation of the concept of integration. General characteristics of recent C^3I prototype KBS developments will be presented, followed by an overview of the C^3I information systems and their constituent DBMSs. Some alternative approaches to KBS-DBMS integration which have thus far been proposed and pursued will also be discussed in terms of their respective merits and viability. The importance of achieving a solution to that problem from a technological as well as an economic viewpoint will also be addressed.

3.1 Current C^3I KBS Prototype Developments

Efforts to apply KBS technology to C^3I functions in recent years have focused on developing prototype KBSs to solve a very specific problem. These prototypes are intended primarily to establish feasibility -- the feasibility of applying KBS technology to perform certain problem solving and decision making aspects of those intelligence functions. Their design and implementation has focused on using AI-based software (e.g., the LISP and PROLOG programming languages) and hardware technology (e.g. LISP machines), rather than on the more conventional hardware and software comprising the C^3I operational computing environments.

For the most part, C^3I KBS prototype development has proceeded in the absence of any considerations for interfacing those systems with or integrating them into the general context of current C^3I-large scale information systems or their constituent DBMSs. Only when a serious effort is made to compare the two system types, does the diversity in their respective problem solving orientations, designs, and implementations become evident. Additional factors such as the respective size of their databases, real-time performance requirements, security, and distributed database and system environments all contribute to that diversity. Thus, efforts to pursue the KBS-DBMS integration problem must necessarily account for this diversity, if they are to be successful.

3.2 C^3I Information Systems and Their Database Management Systems

Today's C^3I computing environments and their DBMSs contrast significantly with the prototype KBSs previously discussed. Many of them consist of conventional large IBM and DEC mainframe systems. Their operating systems and other system software components, e.g., DBMSs, are also rather conventional and may not even reflect the latest state-of-the-art. Application software is typically written in FORTRAN or other common procedural oriented languages. There is little to be found in either hardware or software which has been imported from KBS technology. Consequently, integrating complete KBSs or even portions of their structural and functional elements would be a major design and implementation undertaking in such environments.

The C^3I DBMSs in use today have generally been designed and implemented for large scale system environments. While they may reflect some of the latest DBMS technology, they are not designed to accommodate KBS database elements, e.g., knowledge and rule base elements. Nor are they endowed with facilities to process such elements beyond the typical data management functions of query processing, updating, and deleting. Thus, DBMSs do not contain a rule processing facility, nor do they support the representation of certain kinds of knowledge and information typical of KBSs. Examples of USAF C^{3}I systems and their characteristics are presented in [Lazzara 87]. Just a cursory examination of these fundamental system differences is sufficient to indicate the potential magnitude of any integration effort.

An additional significant difference between the two systems which pertains to the very foundations of any integration effort stems from differences in the kinds of problems each was designed to solve. At this point, it will suffice to point out that DBMSs are designed to facilitate very selective acquisition, modification, or deletion of discrete data elements stored on mass memory devices. KBSs, on the other hand, are oriented towards manipulating large quantities of data in main memory and performing operations on it which extend beyond those of a DBMS. This includes applying rules and rule processing functions, search and control, as well as other knowledge-base processing functions, to data structures, many of which are considerably more complex than those maintained in DBMS databases. These characteristics are at least partly due to the fact that KBSs are not just retrieving data in response to a user command, but are also emulating human cognitive functions such as comparing, combining, analyzing, inferring, and decision making.

At one extreme, the KBS-DBMS integration problem might be perceived as a case of trying to mix apples and oranges, and therefore, is a task with little possibility of success. On the other hand, it might be perceived as achievable, at least to some degree, particularly where there is a real need to have KBSs access DBMS databases to support their knowledge processing functions, as well as to infuse some KBS-type functionality into DBMSs. Some approaches to this integration problem are more viable than others, however, as the next topic will indicate.

3.3 Current Integration Approaches: Merits and Deficiencies

Although their are variations in the literature, see [Jarke 85] and [Missikoff 86], four principal approaches are taken to integrate KBS and DBMS technology: (1) tightly integrate KBS and DBMS functionality to develop an entirely new type of system - a knowledge-based management system (KBMS); (2) augment an existing KBS with DBMS structural and functional elements; (3) infuse KBS functionality into DBMSs; and (4) loosely or tightly couple a distinct KBS to a DBMS.

Thus, the C^{3}I KBS-DBMS integration problem might be approached from any one of these different ways. In the first approach, one could proceed to develop a KBMS. Such a system would constitute an "ultimate" solution to KBS-DBMS integration in that it was developed from the very beginning with KBS-DBMS integration as a goal to be realized in both its design and implementation. But at this point in time, KBMS development is at its most primitive stages. Although a variety of experimental approaches have been tried and others proposed, see [Brodie 86], it is not clear what the architectural features and capabilities of such a system would be. A KBMS has yet to be built, and the likelihood of this kind of system being available for C^{3}I environments in the near term is remote at best. Nor is it clear how and in what way such a system would either replace or augment DBMSs or other systems in those environments. "Deductive databases" represent a variation of this first approach. Within deductive database systems data manipulation functions and deductive functions are merged into a common system. Examples span from the use of Prolog to unify facts and rules with data manipulation functions to object-oriented approaches to capture the notion of active objects. Both approaches possess benefits, as well as drawbacks, and thus, are actively being researched.

The second major KBS-DBMS integration approach, augmenting an existing KBS with DBMS structural and functional elements, is a less ambitious effort, but none-the-less poses formidable difficulties. For instance, considering C^{3}I KBS prototype systems implemented in complex and powerful commercially available expert system shells such as ART and KEE, one can easily envision potential complications. In general, these shells are so specialized in their hardware and software implementations, that engaging in a major effort to evolve DBMS capabilities within them could be a significant undertaking. In addition to the inference and data structuring capabilities offered by these shells, the standard functions of a DBMS would have to developed and made available. Most likely, these functions would be implemented in a language suitable to data management functions that may be different than that of the original programming environment. Thus, in preserving the original programming environment and adding the DBMS functionality, these systems would be challenged by large amounts of code and efficiency difficulties.

The more viable approach, i.e., approach three above, to KBS-DBMS integration is towards expanding DBMSs in the direction of more "intelligent" systems. Because of the substantial investment already made by the USAF in C^3I information system architectures, including recent and ongoing hardware and software updates, and the incorporation of considerable DBMS technology in those systems, the KBS-DBMS integration problem solution seems most likely to be towards of "infusing" KBS functionality into DBMSs. Although conceptually more viable than the other approaches discussed in this section, the infusion of KBS functionality into DBMSs is not without significant technological difficulties. For instance, to provide the functionality of KBSs, e.g., rule processing, and explanation, DBMSs will require the incorporation of conceptual processes, e.g., comparing, abstracting, inferring, and the conceptual definitions and computational primitives to support these processes.

The fourth approach is currently the most common. Two variations exist: loosely coupling a KBS to a DBMS and tightly coupling a KBS to a DBMS. In both arrangements, the KBS and DBMS maintain their identity and their respective functionality. That is, the KBS performs its deductive processes, while the DBMS performs data management functions. A major advantage is that existing databases can be used. In the tightly coupled arrangement, the KBS can request data from the DBMS whenever necessary. When the KBS requires data, it queries the database. In practice, the KBS arrives at a point where external data is necessary to satisfy processing goals. In programming terms, the shell attempts to resolve an external reference, usually by executing a user specified program outside the shell. This program structures the request to a form acceptable by the DBMS's query processor. The query processor is then called with the request. The data manager performs the data retrieval and returns the results to the calling program. The calling program then reformats the results of data retrieved in the form usable by the shell, and the shell continues its deductive processing. As it is evident, many drawbacks exist in a loosely coupled arrangement. The deductive component must state the query in a precise way. This query must be formatted when it is issued and reformatted when data returns. The interface becomes a critical point, often slowing the deductive component. In addition to efficiency and formatting issues, other issues such as dealing with null data exist. In a loosely coupled arrangement, the KBS takes a snapshot of the database prior to performing its deductive processing. That is, selected data is retrieved, structured into the internal format of the KBS, and then processed. Some of the drawbacks that exist in this approach are: the data requirements have to be determined prior to KBS deductive activity; data consistency becomes a problem if the DBMS is updated during the KBS's deductive cycle; and the volume of the data retrieved could be greater than the storage capacity of the deductive component, thus causing severe performance problems.

4 Conceptual Information Processing

It is evident from the preceding section that significant complexities exist in the interconnection or integration of KBS and DBMS technology. Our approach to the integration of KBS and DBMS technology is a variant of case 1 above; that is, the system pursued in our research takes the notion of KBS-DBMS integration as a fundamental goal to be realized in both the design and implementation of the system. Our work has been directed towards defining and implementing a formal systems model for concept-based processing, or in our terminology a **conceptual information processor**. Within this research, we have been proceeding along two dimensions. First, we have been investigating the application of a rigorous semantic theory for describing the conceptual processor system. Second, we have delineated an architecture and conceptual and computational primitives for realizing the architecture. These dimensions will be described after a discussion of concept-based processing.

4.1 An Introduction to Concept-Based Processing

Concept-based processing and derivatives of such processing have been emerging from Sowa's comprehensive and cohesive theory of knowledge representation and computation. Examples of the application of this theory can be found in [Sowa 86], and [Fargues 86]. Sowa has formally described a theory of concept-based processing derived from research in linguistics, psychology, logic, and philosophy. The basis of which is the notion that people understand the world by building mental models. Key assumptions underlying Sowa's theory are: (1) concepts are discrete units; (2) combinations of concepts are not diffuse mixtures, but ordered structures; and (3) only discreet relationships are recorded in concepts; continuous forms must be recorded by patterns of discreet units. Concept types, conceptual relations, and percepts are the building blocks used to represent mental models. These are represented by three types of objects: referents, concept types, and concept relations. A referent denotes individuals, sets, or values explicitly mentioned in the domain being models. Examples of referents are "Bill", [Knowledge Systems Concepts, Datalogics], and 2.17. A concept type asserts the existence of something of a corresponding type, e.g., [PERSON], [BITE], and [APPLE] are concepts. Thus, concept type nodes can represent entities, attributes, states, and events. A conceptual relation denotes how concepts are related or connected, e.g., (AGT) and (OBJ) denote the conceptual relations "agent", "object". Concepts and their conceptual relations are represented by a structure called a conceptual graph. A simple example in Sowa's linear notation of the statement *"Bill bit an apple"* follows:

[PERSON: Bill] <- (AGNT) <- [BITE] -> (OBJ) -> [APPLE]

In this example, concept types are represented as upper-case type labels surrounded by square brackets, and relations are represented as upper-case type labels surrounded by parentheses. Bill is the name of an individual of concept type PERSON. BITE and APPLE are concept types, while (AGNT) and (OBJ) are relations. (AGNT), denoting agent, links an act, in this example "to bite", to the actor Bill. Likewise, (OBJ), denoting object, links an actor to an entity which is acted upon, i.e, (OBJ) links BITE to the APPLE. Thus, the notation represents the statement *"Bill bit the apple"*.

Conceptual graphs and relations are further organized in schemas, where the terminology 'schema' refers to the organization of these building blocks into larger structures. The conceptual graph notation makes it fairly easy to describe schemas and to formally make useful distinctions. For instance, a schema is composed of a coherent set of statements, instead of a series of isolated statements. Likewise, a schema description for a particular concept contains relations to other concepts rather than to slots and values or attributes and values. The notion of schema and the utility of schema as active objects, instead of inert, static descriptions is fundamental to our goal of building a robust functionally integrated knowledge and data system. The conceptual graph formalism provides the representational expressiveness and the computational model to define active objects and conceptual definitions that allow users and applications to interact at a higher level than that of data stored in the database.

4.2 Our R&D Approach and Status

Our current development of an experimental model of a concept-based information processor, i.e., robust KBS-DBMS, has been driven from three key motivations. First, we are aware of the current limitations of both the knowledge base and database system technologies and intend to develop a robust functionally integrated system. Second, we have defined a partial cognitive model which accounts for the types of cognitive-like processes, e.g., abstracting, inferring, comparing, desired in the system. Third, we have been investigating potential infological approaches for performing a rigorous analysis of the concept types, concept relations, and conceptual processes necessary for a particular application. The first motivation has been previously described, the second and third are described as follows.

The cognitive model driving our implementation is based on Sowa's process model for perception. Basically, four functional categories have been described: perception, learning, reasoning, and memory management. Of these four, our concentration has been placed on the later two. To realize an experimental functional capability in these two areas, we have defined an architecture composed of a user interface, reasoning module, information manager executive, and a global knowledge and data base.

Addressing the reasoning module, we have been developing within a Lisp-based environment the conceptual and computational primitives to realize the definition, representation, and manipulation of computer-based concepts. Applying a machine-independent object-oriented paradigm, we have developed data structures for the primary objects of Sowa's theory, and have extended that declarative framework with the inclusion of a procedural component that can be used to represent dynamic real world processes. With regard to the computational analog for memory management, we have designed and are in the process of implementing the information management executive. An architectural scheme has been developed for uniformly storing and retrieving knowledge and data structures, i.e., objects in both memory and disk have identical representation. A significant contribution of this storage architecture is that knowledge structures, as well as data instances, reside on secondary storage and they are accessed and processed as needed.

The infological techniques being explored are derivative from Sundgren's theory of databases, Newell's knowledge level description, and Barwise and Perry's situation semantics. The goal is to furnish the application developer with the logical analysis techniques to define the concepts, conceptual relations, and referents of the application, as well as provide clearly defined semantics for the application. A larger issue in concept-based processing is the role and representation of contextual information and its influence on 'meaning' and 'interpretation'. Although we are not primarily concerned with the interpretation of natural language, we are confronted with the interpretation of concepts in the larger context of individuals, events, location, and time. The theory of situations developed by Barwise and Perry furnishes a basis for these distinctions and the influences of the context on interpretation. The principal idea of their theory is that the meaning of a sentence is a relation between the sentence and the described situations, while the interpretation of the sentence is the described situation. Conceptual graphs seem to be sufficient representations for semantic information, but in addition to computing over conceptual graphs, we are striving for a formal basis for capturing the semantics of domain and the respective contextual influences on interpretation.

5 Summary

In this paper we have characterized some things positively and taken a negative stand with regard to others; in both cases goal has been to promote a correct understanding of the KBS-DBMS integration problem which underlies the difficulty in fusing these technologies. Our attempt has been to "cut" through the plethora of information and research studies that seduces us into failing to recognize the real problem or to believe that simple solutions exist. As an approach to this larger problem, we have described a concept-processing paradigm which we believe will resolve many of the difficulties associated with KBS-DBMS integration. Likewise, we have overviewed the status of our ongoing work in the development of such a system, and also, have delineated some initial goals and approaches for infological analysis.

BIBLIOGRAPHY

[Barwise 83] Barwise, J. and Perry, J., Situations and Attitudes, MIT Press, 1983.

[Bic 86] Bic, L., and Gibert, J., "Learning From AI: New Trends in Database Technology", Computer, IEEE, March 1986.

[Brodie 86] Brodie, M.L. and Mylopoulos, J., On Knowledge Base Management Systems - Integrating Artificial Intelligence and Database Technologies, Springer-Verlag, Inc., New York, 1986.

[Fargues 86] Fargues, J., et al.,"Conceptual Graphs for Semantics and Knowledge Processing", IBM Journal of Research and Development, Vol. 30, No. 1, January 1986.

[Fox 86] Fox, M. and McDermott, J., "The Role of Databases in Knowledge-Based Systems", On Knowledge Base Management Systems, Brodie, M. and Mylopoulos, J. (Eds.), Springer-Verlag, 1986.

[Harmon 86] Harmon, P. (ed.), Expert Systems Strategies, Monthly Newsletter, Cutter Information Corp, August 1986.

[Jarke 85] Jarke, M., et.al., Coupling Expert Systems with Database Management Systems, Artificial Intelligence Applications for Business, Proceedings of the NYU Symposium, Ablex Publishing Corporation, Norwood N.J., 1985, p. 65.

[Kellog 86] Kellog, C., "From Data Management to Knowledge Management", Computer, IEEE, January 1986, p. 75.

[Kerschberg 86] Kerschberg,L. (ed.), Expert Database Systems, Proceedings From the First International Workshop, The Benjamin/Cummings Publishing Company, 1986.

[Lazzara 86] Lazzara, A., et al.,Knowledge Based and Database System Integration: A Baseline for a Design Methodology, Interim Report, RADC-TR-86-132, September 1986.

[Lazzara 87] Lazzara, A., et al.,KBS-DBMS Integration Analysis and Design Alternatives, Interim Report, RADC-TR *in publication*, February 1987.

[Missikoff 86] Missikoff, M. and Wiederhold, G., "Towards A Unified Approach for Expert And Database Systems", in: Kerschberg,L. (ed.), Expert Database Systems, Proceedings From the First International Workshop, The Benjamin/Cummings Publishing Company, 1986.

[Sowa 84] Sowa, J.F., Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, 1984.

[Sowa 86] Sowa, J.F. and Way, E.C.,"Implementing a Semantic Interpreter Using Conceptual Graphs", IBM Journal of Research and Development, Vol. 30, No. 1, January 1986.

[Weiderhold 86] Weiderhold, G., et al., "An Integration of Knowledge and Data Representation", On Knowledge Base Management Systems - Integrating Artificial Intelligence and Database Technologies, Springer-Verlag, Inc., New York, 1986.