

Knowledge Based System and Database Management System: An Integrative Framework

G. Premkumar
Iowa State University

Knowledge based management systems has gained significant importance in the last few years, primarily from the enhanced capabilities achieved through integration of the two technologies - artificial intelligence and database management systems. This paper develops a framework for integrating the two technologies. The three dimensional framework links the various knowledge representation schemes with the data models used in databases and the architectures used for linking the knowledge base with the database system. This framework facilitates in communicating the latest developments in the field of knowledge based management systems and also aids the designer in making the appropriate choice from the various options available within the two technologies.

The research systems in the field are mapped onto the framework. An analysis of the mapping reveals that some research areas are more popular than others. Rule-based knowledge representation with loose coupling to relational databases are found to be popular for integrating the two systems. Integrated solutions with object-oriented formalisms are also becoming common in recent years.

Potential areas for future research are identified. Also, the implications of the research for the practitioner and the strategies for commercial exploitation of KBMS are discussed.

The integration of artificial intelligence (AI) and database technologies has become an important stream of research in the two fields (Brodie & Mylopoulos, 1986; Kerschberg, 1986). A new class of information systems,

capable of powerful and efficient knowledge based processing, called Knowledge Based Management System (KBMS) has become popular as a result of merger of these two technologies. KBMS is a system providing highly effective management of large shared knowledge bases for knowledge directed systems (Brodie, 1986).

A major reason for the integration of these two technologies is the realization that these are complementary technologies with the potential for enhancing the capabilities of AI systems with database features, and database management systems (DBMS) with learning and other features found in AI systems. On the database side, the ability to provide functionality in information systems using deductive, inductive and plausible reasoning, knowledge representation, heuristic search, knowledge validation and refinement provides opportunities to extend the scope and use of database management systems. On the AI side, the use of database features such as query optimization, concurrency, integrity constraints, data security, and error recovery, provide the facility to efficiently store and use the knowledge that is required for AI systems.

There have been criticisms in recent times of the failure of knowledge based systems (KBS)/ expert systems delivering their promised potential and the slow growth in the use of these systems in actual practice. "Are these systems merely an intellectual exercise for the academic community with only marginal utility value for practitioners?" has been a major question facing the practitioner. Although many expert system shells and software are available in the market, users are yet to use these in a significant way in their

IS applications. The reasons for these are slowly emerging. Most expert systems require rules (or a knowledge base) and facts (or a database) to be integrated to make an application useful in real-world. Success in the use of expert systems is hinged on effectively linking the KBS with the firm's DBMS, that provides the "facts" component of the system and which is already existing in the organization. It is only recently that major efforts are focused on this integration, both in research and in practice. This paper proposes a framework for integrating the two technologies, DBMS and AI, and identifies the present status, problems, research issues, and future growth directions for this area.

Need for Integration

A major reason for the integration of KBS and DBMS is the significant enhancements brought to the individual subsystems by utilizing some of the features of one in the other. A brief description of the enhancements are discussed below.

DBMS Enhancement

Knowledge based systems can enhance the capabilities of the DBMS by providing:

- a) Intelligent interface to database,
- b) Natural language interface to database,
- c) Query optimization,
- d) Database maintenance, and
- e) Data model development.

Intelligent interface to DBMS typically involves providing reasoning capabilities in query processing, and higher level query languages that allow for more powerful and complex operations with minimal programming skills. These interfaces basically simplify and improve the user interface to DBMS.

The natural language interface typically contains a reasoning front-end that accepts queries in natural language dialect, maps the query into a logical form, performs deduction, and then interacts with the DBMS to retrieve the information. KM-1 (Kellog, 1982) is a typical example of a natural language interface which uses a English-like dialect to interact with a front end AI machine that transforms the query and then retrieves the information from a relational database.

Query optimization using deductive reasoning (Reiter, 1978; Grishman, 1978), optimization of multiple queries and minimizing disk access, development of efficient access routes using historical data (Grant & Minker, 1981), improvement of response time for simple and repetitive queries by using an abstract of the database (Rowe, 1983) are but few of the uses of AI methods for efficient query processing.

Data maintenance operations such as integrity checking and data consistency maintenance can be supported using a knowledge based system. A rule based AI system (Goldstein & Bobrow, 1980) or a deductive logic based system (Nicolos and Yazdanian, 1978) is frequently used for checking and maintaining data integrity in the database. Another useful feature is the monitoring of temporal conditions in a database and taking appropriate actions such as updating the database or generating reports for users, whenever conditions are satisfied. Stonebraker (1982) describes a rule based system (RAISIN) which has some of these features incorporated in the query processing component of a relational DBMS.

The need to integrate the two systems has resulted in significant progress in the development of new data models that can capture the richness of knowledge representation required in AI systems. The limitations of the relational model have led to the development of the semantic data model (Hammer & McLeod, 1981; King & McLeod, 1985), functional data model (Shipman, 1981), and object oriented data model (Copeland & Maier, 1984). These models capture the information richness of the real-world, and thereby provide a better representation of the real world.

Enhancement of Knowledge Based Systems

A simple knowledge based system stores most of the relevant data in main memory while processing its applications. However, as these systems grow in size and the expertise domain expands, it becomes necessary to store the data in a secondary storage such as an external DBMS. The facilities provided by the DBMS in terms of data management, concurrent access, and query optimization provide robustness for the KBS to handle large volumes of knowledge. Also, this provides flexibility for the KBS to access a central DBMS that is used for other applications in the organization. Steinberg (1990) describes an expert system in American Express that provides credit authorization by using a corporate customer database that is used for various other applications.

Deductive databases with enhancements to access an external device are common examples of extension of KBS capabilities using the DBMS features. Parsaye (1983) discusses the use of database concepts such as schemas, functional dependencies, integrity constraints etc., in a Prolog based system. Carey, Dewitt, and Graefe (1986) discuss the use of concurrency control and error recovery mechanism for a Prolog based system, using a two-phase locking concept, that is very similar to mechanisms used in DBMS. STROBE (Lafue & Smith, 1986) is a knowledge based system where the database concepts of semantic integrity checking, and file management are integrated into an object oriented environment. Dahl (1982), and Lafue (1983) also provide examples of KBS where the capabilities have been enhanced using DBMS concepts.

The Framework

Having highlighted the need to integrate the two technologies, we provide below a framework for linking the two technologies. This three dimensional framework links the various knowledge representation schemes with the database models and the architectures used for linking the KBS with the DBMS.

Objectives of the Framework

A framework helps in gaining a better perspective of the field by providing a structure for understanding the status of research in the field and identifying the future research issues and directions. The proposed framework has two specific objectives:

- a) To facilitate communication among researchers and practitioner about the developments in knowledge based management systems; and
- b) To aid the designer of KBMS in the selection of appropriate database tools and knowledge representation schemes for the design of the system.

The primary purpose of a framework is to facilitate communication among researchers, an essential element of scientific progress. The communication helps in consolidating the present knowledge about a particular subject and “agreeing on what we know”. It is useful for identifying areas with research potential and areas where progress has been slow. It is also useful in identifying the impact of various scientific and technological advances on the subject areas. In MIS, early research frameworks of Gorry & Scott Morton (1971), Mason & Mitroff (1973), and Ives, Hamilton & Davis (1980) have been very useful in consolidating past research and directing future research efforts.

A framework also helps in identifying design options, providing design guidelines, and highlighting the issues to be addressed in the design process. The early MIS frameworks have been useful in identifying the design guidelines and issues in the development of information systems under various decision making conditions.

The Integrative Framework

The framework illustrated in Figure 1 focuses on the underlying scientific disciplines of KBMS. The three dimensions of the framework are:

- a) Knowledge representation schemes
- b) Data models
- c) Architectures for integration.

Knowledge representation is the process of representing facts and relations between facts in a manner that is ame-

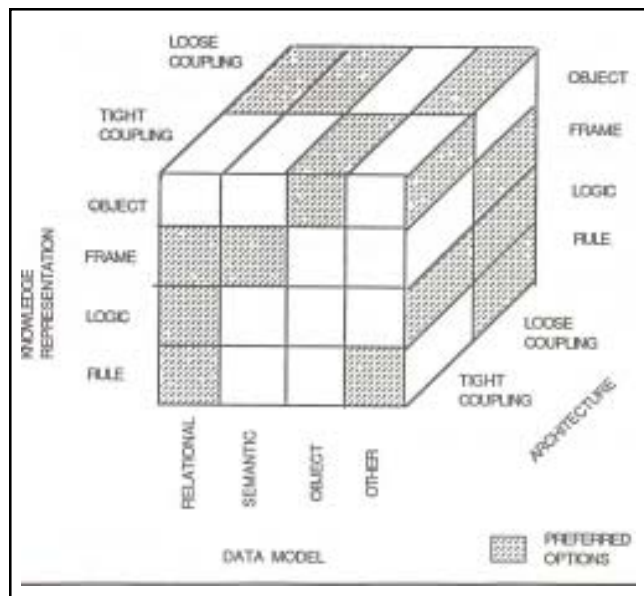


Figure 1: Framework

nable for easy access and manipulation. The common knowledge representation schemes that are used are (Brodie & Mylopoulos, 1986b):

- a) Rule-based representation
- b) Logic-based representation
- c) Frame-based representation
- d) Object oriented representation.

A data model is a mechanism for specifying the structure of a database and the atomic operations that may be performed on the data in that database. They have a set of constructs to define the data stored in the database (DDL) and a manipulation language (DML) to update or manipulate the data. The data models that are in common use (Date, 1990) are:

- a) Hierarchical model
- b) Network model
- c) Relational model
- d) Semantic model
- e) Object oriented model

Hierarchical and network models are frequently considered as a single class of data models since they are based on similar modeling concepts (King & McLeod, 1985). Also, there is very little research in the KBMS field relating to these two models to actually differentiate them. Hence, we have chosen to consider them as a single category in our framework.

There are various design options for integrating database systems with knowledge based systems. The choice of a particular option is dependent on the problem addressed, the

data model and knowledge representation scheme currently being used, and other operational constraints. Various architectures have been developed for the integration of KBS and DBMS based on data volume, reasoning complexity, data integrity, data protection, data volatility, and data source.

Fishman (1986) identifies four basic approaches to integration of the two systems - KBS-DBMS tight coupling, KBS-DBMS loose coupling, KBS within DBMS, and DBMS within KBS. In the first two approaches, the two subsystems are individual entities with a communication channel, while in the last two, one of the subsystem dominates the other. In the last five years the distinction of the last two categories from the first two have eroded and researchers are increasingly referring to the first two categories, tight and loose coupling, as the primary coupling architectures (Choobineh and Sen, 1987; Stonebraker and Hearst, 1988; Brodie and Mylopoulos, 1986). Hence, the two architectures considered in our framework are:

- a) KBS-DBMS loose coupling
- b) KBS-DBMS tight coupling

We have four options in data model, four options in knowledge representation and two options in architecture resulting in a total of thirty two different design options for integrating the two systems. The selection of a particular option is dependent on various design parameters and operational considerations such as:

- a) Requirements of the application - access requirements, data location and structure, problem representation, data security, and data integrity;
- b) Compatibility with existing DBMS if access to DBMS is required;
- c) Compatibility with existing software, since some software support only a few knowledge representation schemes;
- d) State of the art in technology along all the three dimensions;
- e) Future growth potential for the system - Growth in terms of data volume, complexity of logic and processing; and
- f) Technical skills of the software personnel.

A brief description of the options along the three dimensions is given below, followed by a mapping of research systems on to the framework and finally an evaluation of the framework.

Data Models Used in Database Systems

Hierarchical and Network Data model: Hierarchical data model is a collection of record types and link types with the constraint that each instance of a record type in a given tree must have exactly one parent record of the parent type. Network database enlarges the flexibility by allowing multiple link types between record types and a given record instance having multiple parent record instances of a given record type. Thus many to one and many to many relationships are easily represented. The main approach of hierarchical and network models is to model data using records and inter-record connections (links), which provides the semantic connection and a physical access path to the record. A major limitation of these models is the direct correspondence between physical access path and the logical inter-record links and the need for the user to 'navigate' to reach a record.

Relational Model: A relational model consists of a number of (n-ary) relations and a collection of underlying domains. The interconnection between records is achieved implicitly through a common field value. Hence, a relational data model provides greater flexibility as the user can determine dynamically the logical structure subject to the constraints of normalization. The shortcomings of the relational model are:

- a) Fixed data types and inability to define new abstract data type;
- b) Lack of differentiation between type definition and data declaration;
- c) Lack of ability to portray variety and complexity of real data due to structural limitations;
- d) Lack of modeling power due to necessity to simplify real world data structure, thereby losing entity identity;
- e) Lack of facility to store temporal data in the same DBMS;
- f) Lack of extensive data manipulation facility in the DML of DBMS, leading to use of other higher level languages like C, and Cobol for extensive data manipulation.

Semantic data model: These models provide a richer set of modeling constructs, which is closer to the way users think about entities. The primary impetus for generating interest in semantic data models is from the rich knowledge representation schemes used in artificial intelligence. Semantic

networks are used effectively to represent large amounts of abstract domain knowledge by employing a network of objects (nodes) connected by relations (directed edges) which could be of different types. King & Mcleod (1985) describe three generic kinds of semantic relationships that should be expressed in a database as:

- a) "has-subtype" relationship, which logically links an object type with another object type that is a subtype of the former;
- b) "has-attribute" relationship, which connects an object with another object that describes some aspect of the first;
- c) "has-instance" relationship, which links a type to an object that is an instance of that type.

Early researchers have attempted, with only limited success, to map semantic constructs on to existing relational model. Codd (1979) presented an Extended Relational Model in which a type is represented by a relation (entity relation) that contains a single column that specifies the "surrogate" (internal unique identifier) for every instance of the type. Weiderhold & El-Masri (1979) developed an extension to a relational model which provides some facilities of a semantic data model. Researchers have also developed their own semantic models, with built-in data definition and manipulation language (See Peckham & Maryanski (1988) for a review).

Object oriented data model: Object oriented data model is an extension of the semantic data model with more advanced features. It defines a collection of object classes and specifies structural relationship among object classes and the operations that access and manipulate each object class (Parsaye, 1989). The data model essentially consists of four constructs: object, method, message, and class. An object accepts messages that requests it to access, modify or return a portion of its private memory that consists of a list of numbered instance of variables. A group of structurally similar objects that respond to the same set of messages is a class. A class contains the procedures/methods that its objects use to respond to messages. Classes are organized in a hierarchy, so that they can share common structures and methods in a superclass. Objects communicate and perform computations via messages. A class of models called functional data models, as described by Shipman (1986) in the DAPLEX system can be considered as precursors to object oriented data modeling. Copeland and Maier (1984) have incorporated most features of an object oriented data model in their system "Smalltalk".

The object oriented framework provides better support for managing time and changes in databases. The first advantage of 'referential transparency' is achieved since any change in entity value is automatically seen by all entities which refer to it, unlike in relational model, where

there is no facility to automatically propagate the value. The second advantage of 'version management', enables storing old versions of objects with time stamps as a unique identifier.

Knowledge Representation in Artificial Intelligence

Rule-based Representation: Rule-based or procedural representation views the knowledge base as a collection of processes. Production systems, an example of such a scheme, consists of a set of if-then rules or inference engine, a collection of facts or knowledge base, and a control strategy that specifies the order in which the rules will be instantiated and the means to resolve conflicts when multiple rules are matched simultaneously. Content reference and meta-rules (Davis 1980), factoring (Woods 1986) and various other procedures are used to improve the computational efficiency of the system. Hayes-Roth (1985) describes some of key properties of rule-based knowledge representation. Many specialized programming languages such as OPS5 have been developed to incorporate this representation scheme. Many early systems such as XCON, MYCIN, and PROSPECTOR were developed as rule-based systems.

Logic representation: Logic representation employs the notion of constant, variable, predicate, logical connective and quantifier to represent facts as formulas. A database is a collection of logical formulas that provide a description of a state. Addition or deletion of such formulas result in modification to the database. A logic program defines a procedure to answer a query through a set of formulas or Horn clauses which are of the form; $Y \leftarrow X_1, X_2, X_3 \dots X_n$.

Facts, intentional rules specifying relationships, integrity constraints, and control constructs are all expressed in the same way uniformly. Such a representation provides deductive reasoning. Prolog which is primarily used in logic programming, also incorporates extra-logical features to improve processing efficiency. Genereseth and Ginsberg (1985) provide an overview of logic representation and logic programming. Research efforts on using logic have focused on representing temporal information, beliefs, defaults, and incomplete knowledge.

Frame-based Representation: The rich representation mechanisms and semantics used in frame-based representation has made it a very effective and popular knowledge representation mechanism (Fikes and Kehler, 1985). A frame based system consists of a semantic network in which each node is a frame, which contains a collection of information stored in different slots, corresponding to the concept, its attributes, default values, and the actions to be taken (Minsky, 1975). In addition to encoding and storing

beliefs and data about a problem domain, frame based representation performs a set of inferences that extends the explicitly held set of beliefs to a large, virtual set of beliefs. Information is also shared among multiple frames by the property of inheritance. Various languages such as KEE (Kehler & Clemenson, 1984), SRL+ (Fox, 1986), and KRL (Bobrow and Winograd, 1977) have been developed to manipulate the frame based representation.

Object Oriented Representation: Object oriented representation is an extension of the concepts developed in frame-based representation. It attaches declarative representations of knowledge to objects. The objects can be entities or relationships between entities. The application oriented behavior is encapsulated into abstract operation types which are themselves objects. For example, an operation called 'Have-lunch' might model an aggregation of component objects such as persons, place, time, cost, together with the specified constraints on food. Each instance of 'have-lunch' will be represented as an object.

The constructs used in object oriented data modeling — object, method, message, and class — provide the foundation for object oriented knowledge representation. Very often systems are designed where object oriented representation is implemented in an object oriented data model. The object oriented approach provides portability as evidenced by its adoptions in different programming languages such as C++, LOOPS, ACTORS, and FLAVORS (extensions of LISP).

Architectures for Integration of KBS and DBMS

KBS - DBMS Loose coupling: This approach is an attractive way of increasing the utility of an existing DBMS through enhancing its capability using a KBS or linking a KBS to a large DBMS, thereby providing an extensive knowledge base for the KBS system. The coupling process consists of a precompilation mechanism in the KBS system, where requests for data are collected as a part of the deduction process. The collected database calls are optimized based on past history, recognition of common sub-expressions in the current queries and simplification. The optimized query is then translated into DBMS query language, executed by the DBMS, and the answer is sent to KBS.

Typically, commercial applications are developed using an expert system shell with an external interface to a DBMS. The advantages with this approach are the existence of many expert system shells with loose couplings to one or more DBMS, the fairly simple and flexible approach to integrate the two systems, use of existing databases without affecting other database applications, and the limited amount of reprogramming or change, which is a significant inhibitor in

most organizations. However, the major disadvantages of the approach are the volatility of the rule-base which is memory resident, reduced flexibility, inability to share data, data integrity problems in using a "snapshot" of data from a dynamic database, inefficient query processing, and error recovery problems. Hence, this architecture is normally used when:

- a) A 'static' data retrieval is sufficient,
- b) KBMS is not very sensitive to changes in the data in DBMS,
- c) the data in DBMS is relatively stable,
- d) simple interface between the subsystems is sufficient,
- e) quick and easy implementation on an existing DBMS is necessary,
- f) a logically rich and complex analysis of information using a KBS in an off-line mode is necessary,
- g) no updates are required on the DBMS, and
- h) data integrity is not a major concern.

Loose coupling, although simple and convenient, unfortunately, compromises on the basic strengths of databases such as concurrency, query optimization, data integrity, and error recovery.

KBS - DBMS Tight coupling: The major difference between the loose and tight coupling approach is that in loose coupling, the data is retrieved from DBMS through a communication channel and stored as a "snapshot" in the KBS database and hence is static, while tight coupling provides dynamic data extraction (Jarke & Vasilliou, 1986). Tight coupling between KBS and DBMS supports dynamic access and the communication channel is open during the KBS operation. It is normally implemented when any of the following conditions exist:

- a) there is a high degree of volatility in the data stored in DBMS,
- b) there is a necessity for 'dynamic' retrieval of data,
- c) the KBMS is very sensitive to changes in the data stored in DBMS,
- d) KBMS is one of multiple users of the DBMS,
- e) data integrity and data security considerations are very important,
- f) underlying data structure in DBMS is very complex for a simple interface,
- g) there is substantial gains in efficiency by integrating the two subsystems.

Mapping of Existing Systems in the Framework

Many research systems have been implemented in the last ten years that have used the various options in knowledge representation, data models, and architecture. A map-

ping of the various research systems developed so far onto the framework would help us in identifying the relative merits of the different approaches and the range of considerations that determine the final choice. This would also assist in providing practical guidelines for expert systems development.

The mapping of the different research systems, based on the limited information available in the research literature, is given in Table 1. In order to link the analysis to availability of commercial systems, Table 2 provides some of the features of popular expert system shells that are available in the market. Table 1 presents the 3-dimensional framework in a tabular form with four sections representing the four knowledge representation schemes. The systems in each knowledge representation schemes are mapped along the other two axes - data models and architectural options.

A cursory analysis of the table indicates that a few cells in the matrix are more popular compared to others because of the potential opportunities available in exploiting the benefits of integration. Loose coupling seems to be the most popular approach as can be seen from the number of systems listed in that category. Also, Table 2 reveals that loose coupling is the most popular option among the commercial ES tools. We also notice a significant increase in the number of object-oriented systems being developed in the last five years. This clustering of research systems indicates that with the available technologies certain combination of options are more feasible compared to others. We will briefly examine the systems in each cell to better understand the clustering and its implications.

Using Rule-based Representation

LISP or LISP based languages such as OPS5 or OPS83 have been mainly used in systems using rule-based representation. Any system using this representation requires capability in database language to provide the generalized "if-then" rule constructs or another language with such capability and ability to be linked to the database language.

Tight coupling between DBMS and KBS using rule-based representation have been primarily achieved by having the rulebase within the DBMS and controlled by the DBMS. POSTGRES (Stonebraker & Hearst, 1988) is a typical example of this category of systems. It uses OPS5 like rules and Ingres-type DBMS to enhance the capability of the DBMS by having events (e.g. update or retrieve) trigger the rules to carry out suitable updates on other parts of the database. We are also seeing a category of integrated systems with tight coupling features in the object oriented environment. The methods and procedures for the object class are stored using a rule-based representation. McCarthy and Dayal (1989) describe a system (HIPAC) where each of the objects have rules that are triggered based on a event-condition-action model. It uses a object oriented data model and is implemented in Smalltalk-80.

Loose coupling with rule-based schemes are quite popular. Vesonder et al., (1983) describe a system "ACE" (Automated cable expertise) which uses the cable reports stored in a conventional DBMS to perform complex analysis and produce various maintenance reports. Olson & Ellis (1982) describe a system 'Probwel' which does analysis of problems in oilwells using historical data stored in a conventional DBMS. We observe that in the above two systems, the KBS does the analysis in an off-line mode, accessing the database to only retrieve information. KM-1 provides a natural language interface using a front-end AI machine that transforms the query and then retrieves the information from a relational database (Kellog, 1982). Various expert system shells, which use rule-based representation, provide a loosely coupled interface to external DBMS.

KBS based on rule-based representation have been used within DBMS for enhancing the effectiveness and features of the DBMS. RAISIN uses a rule based representation to enhance the capabilities of a relational DBMS (INGRES), using backward chaining to answer queries in QUEL and forward chaining to propagate database updates to dependent data elements (Stonebraker, 1982). Kung (1986) describes a system where heuristic search algorithms have been incorporated in QUEL to enable it to select from a group of search algorithms based on its own past search performance data. Similarly, database capabilities such as integrity constraints, view and protection services are being addressed using rule based approaches (Jarke et al., 1982). These systems are primarily focused on improving the capabilities of existing DBMS by providing a KBS within the DBMS.

Using Logic Representation

Deductive database systems, using logic representation and programming, have become very popular in expert systems (Gallaire, Minker, & Nicolas, 1987). Tight coupling is feasible between logic based system and relational database systems, as the latter have been articulated in terms of logic formalisms. In the approach described by Naqvi (1986) and Chakravarthy, Fishman, & Minker (1986) the system is organized in two parts; extensional database (EDB) and intensional database (IDB). All factual data, expressed as variable free formulas (ground formulas) are kept in EDB, while non-ground formulas remain in IDB. This improves the efficiency, as the deduction process is localized in IDB while data retrieval and updating are limited to EDB.

The fifth generation project (FCGS) in Japan also uses a tight coupling approach. Kunifuji and Yakota (1982) describe a tightly coupled system for query processing where the Prolog program simulates a Prolog inference engine; it skips all database predicates until all non-database predicates are eliminated, and after the metalevel inference, the queries are sent to the DBMS and the values returned are

	Relational	Semantic	Object-oriented	Others
s t	Tight Coupling	Postgres (Stonebraker & Hearst, 1988)	HIPAC (McCarthy & Dayal, 1989) OSAM (Raschid & Su, 1988)	
	Loose Coupling	KM-1 (Kellog, 1982) RAISIN (Stonebraker, 1982) Jarke et al., (1982) Kung (1986)		ACE (Vesonder et. al, 1983) PROBWELL (Olson & Ellis, 1982)
n o	Tight Coupling	BERMUDA (IOANNIDES, 1988) EDUCE (Bocca, 1986) Naqvi (1986) Kunifuji & Yakota (1982)		
	Loose Coupling	EDUCE (Bocca, 1986) PROSQL (Chang & Walker, 1986) Warren, 1986; Carey et. al, 1986 JCV (Jarke, 1984); Sciore & Warren, 1986; Parsaye, 1983; CGW (Cerei, 1986)		
s a	Tight Coupling	SRL+ (Fox, 1986)	PRISM (Shepherd & Kerschberg, 1984)	
	Loose Coupling	SRL (Fox, 1986) IRUS (Bates, 1986) Moser, 1984 Bea, 1983 Spark-Jones, 1982		
a t	Tight Coupling		HIPAC (McCarthy & Dayal, 1989) GEMSTONE (Copeland, 1984) Raschid & Su (1988) STROBE (Lafue & Smith, 1986)	
	Loose Coupling	GEM (Tsur, 1984)		

Table 1: Mapping of Research Systems to the Framework

System Name	Knowledge Representative	Coupling Option	Company Name
ART	Frame	Loose	Inference Corp., CA
KEE	Frame	Loose	Intellicorp, CA
S1	Frame	Loose	Teknowledge, CA
Knowledge Craft	Frame	Loose	Carnegie Group, PA
Personal Consultant+	Frame	Loose	Texas Instruments, TX
ESE	Rules (EMYCIN)	dBASE III Loose (IBM Mainframe)	IBM, CA
OPS5	Rules	Loose	Computer Thought Corp., TX
KES	Rules	Loose	Software Arch. & Engg., MA
M1	Rules	Loose (PC based)	Teknowledge, CA
EXSYS	Rules	Loose	EXSYS, NM
Insight 2+	Rules	Loose (dBASE III)	Level Five Research, FL
TIMM	Rules	Loose	General Research Corp., VA
NEXPERT OBJECT	Object	Loose	Neuron Data Corp., CA
ENVISAGE	Logic (Prolog)	Loose	Systems Design Software, MA
ESP Advisor	Logic	Loose	Expert Systems Intern'l, PA
ADS	Rules	Loose	Aion Corporation

Table 2: Expert System Shells

used to construct Prolog answers. Dahl (1982) proposes a total logic based system wherein user's queries are translated into an internal representation, an improved version of logic, and then processed.

EDUCE (Bocca, 1986) provides both tight and loose coupling of a Prolog interface to INGRES. New prolog predicates are used that take QUEL commands in various forms as arguments. BERMUDA also provides both tight and loose coupling and has features very similar to the EDUCE system (Ioannidis, Chen, Friedman, and Tsangaris, 1988). A unique feature in this system is the buffering mechanism that is able to cache and reuse query answers. Other interesting features of the system are the transparency of the database to the user, minimizing the number of queries, and the operation of multiple Prolog processes.

Loose coupling between Prolog and relational systems was extensively used in the early developmental stage when Prolog was interfaced as a front-end to relational DBMS (Jarke, Clifford, & Vassiliou, 1984; Chang & Walker, 1986; Ceri, Gottlob, & Widerhold, 1986; Bocca, 1986). In Jarke's (1984) system, Prolog front end is coupled to SQL through an intermediate language (DBCL) which is a variable free subset of Prolog. User's queries in Prolog are translated into an intermediate language, optimized syntactically and semantically, and then translated into the target language. PROSQL was developed in IBM to interface Prolog with SQL/DS (Chang and Walker, 1986). All data returned from the database is loaded into Prolog, which uses the data in its programs. Prolog and SQL run on independent machines and communication between them is achieved by sending messages. The user interface to DBMS is not transparent, and there are memory limitations due to retrieving all the data in a single access.

Ceri et al., (1986) describe a system (CGW) in which Prolog requests for access to database predicates, and the query answers returned from the DBMS are asserted into the Prolog program. An important feature of this system is that it keeps track of all the query answers and any new query is checked with respect to existing query information and an access to database is made only if it is not already available. EDUCE (Bocca, 1986) and BERMUDA (Ioannidis et al., 1988) also provide loose coupling of Prolog to DBMS. Commercial Prolog systems such as Quintus (1987) and BIM (1988) provide interfaces with popular relational database systems such as Sununify and Oracle. A major problem of loose coupling in deductive database systems is the mismatch between tuples and sets. While logic (or Prolog) answers queries at the rate of one tuple at a time, DBMS retrieves data in sets of tuples. The retrieved data has to be cached in temporary memory till it is processed by the logic based KBS.

KBS are used within DBMS primarily for query optimization. In conventional query optimization the query is optimized using the syntactic information. Newer techniques using semantic query optimization (Jarke 1986,

Chakravarthy et al., 1986) make use of KBS; it uses heuristics and semantic integrity constraints of the database system to decrease query evaluation costs. Ullman (1986) presents an approach for query optimization using logic and 'capture rules'.

The effectiveness of logic based KBS can be enhanced by incorporating database features such as query optimization, concurrency control and data integrity in their systems. Warren (1986) proposes dynamic indexing and data persistence in Prolog, and improving efficiency by preoptimizing queries through compilation; but this imposes a limitation on the user to state in advance the different kinds of expected queries. Sciore and Warren (1986) focus on providing three features; storing the tuples in secondary storage in a format facilitating efficient storage, converting the tuples to internal format of Prolog at the time of processing, and buffer management. Carey et al., (1986) describe the concurrency control and error recovery mechanisms that can be incorporated in Prolog. Parsaye (1983) discusses the design issues in developing systems in Prolog and shows how schemas, functional dependencies and integrity constraints can be efficiently expressed in the language.

Using Frame Representation

Tight coupling of DBMS with frame-based KBS have not been very successful. Fox (1986) proposes a system SRL+ which could fit into this category.

In the category of loose coupling, the focus has been to provide natural language front-end to DBMS by using frame based representations for processing natural language statements. Bates et al., (1986) presents a system called IRUS which acts as a front-end to a relational database system (System 1022) with a conceptual modeling layer in between them. Various other frame based systems that provide natural language interface to DBMS have been discussed in research literature (Moser, 1984; Spark-Jones and Bogonaev, 1982). In another approach, SRL (Fox, 1986), a frame based language has been coupled to a database providing two-key access, context and schema name. The coupling between the two systems is via transactions, with the most recently accessed schemata being kept in LISP environment for quick access. Many systems such as CELLISTO, INET, ISIS, and PDS (Fox, 1986) have been built in this environment. Fikes & Kehler (1985) describe a system called STAR-PLAN that is used as an intelligent aid in the diagnosis and correction of satellite malfunctions.

Typically Frame-based systems do not provide direct facilities for describing how the knowledge stored in frames is to be used. This is best done using procedural or rule-based representation. A great deal of success has been achieved recently by using hybrid systems, where rules are embedded within the frames. For example, COMPASS (Prerau, 1985) uses rules within frames for representing the knowledge of telephone maintenance operations. The KEE system, a popular frame-based environment for develop-

ment of systems, also provides access to a relational DBMS.

Using Object Representation

GEMSTONE (Copeland & Maier, 1984) is an example of an object oriented system where tight integration has been provided. GEMSTONE has been built by integrating a set-theoretic data model and SMALLTALK language; the latter provides an object-oriented environment. The set-theoretic model provides features for structured type definitions, declarative syntax, and representation for time varying data and thus enables to overcome the limitations of relational database in these areas. It also provides a single language for data retrieval, update, general computations and operating system commands. The front-end called 'executor' interacts with multiple users and the back-end called 'object-manager' handles operations of concurrency control and secondary management. Issues of maintaining database consistency and protection are yet to be fully resolved.

STROBE (Structured Object Knowledge) is an object oriented system with database features (Lafue & Smith, 1986). The original interlisp environment has been extended in two ways; object oriented modeling features and database concepts of secondary storage, semantic integrity management and file management have been incorporated. Query optimization on disk based files, indexing on slots, handling arbitrary grouping of fields, using a single language (interlisp) for interacting with the system are some of the issues that is addressed by the system.

HIPAC (McCarthy & Dayal, 1989) is an object oriented data base management system, based on Smalltalk-80. It uses an extended data model called PROBE (Dayal & Smith, 1986). Actions are triggered by event-condition-action rules. Raschid & Su (1988) describe an object oriented representation and data model, where facts and rules are integrated within the object classes. OPS5 like production system is used for applying rules while processing transactions in the KBMS.

GEM (Tsur & Zaniolo, 1984) is a object oriented model loosely coupled to INGRES database. The front-end interface provides the object oriented environment and the coupling is achieved by mapping the object features to data structures in QUEL through intermediate parse trees. The system performance has not been adequate for executing object oriented retrieval and also the communication loads have been high.

Analysis of the Framework

A critical examination of the mapping in Table-1 clearly reveals that some cells in the framework are more popular compared to others. Various reasons can be attributed to this clustering - compatibility and availability of the technologies, requirements of the target application, availability of technical skills, and problems in integration. We will briefly

review some of the research issues that arise from the framework.

Loose coupling seems to be the most popular approach to link KBS to DBMS. It provides the practical benefits of linking KBS to existing databases in an organization without replicating large volumes of data. Also, the availability of many expert system shells with loose interface to DBMS, as is evident from the data in Table-2, makes this approach very attractive. However, it seems to be only a near-term practical solution as the integration is not very efficient and optimization facilities are limited. Tight coupling, on the other hand, exhibits significant advantages over loose coupling, but most of the systems are still in research and prototype stage and yet to be commercially exploited. The significant progress in research, especially in object oriented and Prolog based systems, indicates that these will translate into practical systems in the near future.

Rule based approaches seem to be ideal for small systems and are easier to integrate with commercial DBMS, which are not necessarily relational. It is also easier to design with existing skills because the constructs used in rule-based languages are similar to the constructs used in many of the popular programming languages such as Fortran and Cobol. As one of the first representation schemes used in AI, many commercial systems are available with this scheme. Since production systems have proven to be successful in industrial environments, development of tightly coupled production systems presents a potential opportunity for further research.

Logic-based representation seem to be ideal for interfacing with relational DBMS because similar logical formalisms are used in both the models. Hence, both tight and loose coupling of DBMS to Prolog based systems have attracted much research. It also seems to be useful for incremental approaches - for example, adding additional features such as query optimization in DBMS, or augmenting Prolog based systems with database features such as integrity checking, functional dependencies, and error recovery. More applied research is useful in this area to translate the ideas developed in the research systems into commercial applications.

Frame-based representation provides a rich set of modeling constructs for describing a real world situation. However, because of its complex data structures, direct interface to existing DBMS is a serious design problem. Some commercial systems such as KEE do provide limited interface to relational DBMS. Hybrid representation is common with this scheme, since the procedural component of the knowledge has necessarily to be captured using a rule based component in each of the frames. Object oriented systems, an extension of this scheme, seems to attract more research attention in recent times (Parsaye, 1989).

Loose coupling in object oriented systems is not very popular because of the difficulty in integrating complex knowledge representation scheme with the available database structures. A one-to-one coupling between object ori-

ented data model and object oriented representation seems to be ideal for systems developed using this scheme. The delineation between knowledge base and database is not very clear in this scheme since most systems are developed in a single programming environment such as Smalltalk or C++.

Each scheme has its own inherent advantages and disadvantages and the selection of a particular option is dependent on the problem in hand and the constraints imposed by the problem context. Recently, there have been promising results in using mixed representation in complex systems. For example, in object-oriented systems, procedures and methods are incorporated using rule based representation in the object-class and are triggered by certain events such as update or delete. Further research may identify optimal combinations that will be effective in various situations.

Evaluation of the Framework

A framework to be accepted by both practitioners and researchers as a valid and useful framework has to exhibit certain features. Some of the important features that need to be exhibited by a framework are (Lersch & Mantei, 1984):

- a) **Acceptance** - The framework has to be accepted by both practitioners and researchers. While researchers require the framework to be based on sound theory and have a scientific consensus on the dimensions, the practitioners are interested in it having practical utility and enable them to better understand the field.
- b) **Precision** - The dimensions used in the framework should be unambiguous, ie. the axis should be orthogonal and there should not be any redundancy in the cells.
- c) **Generality** - The framework has to be comprehensive and be capable of categorizing all types of knowledge based management systems presently available and those that will be developed in the near future.
- d) **Parsimony** - As much as the framework needs to be comprehensive, it should also be simple, parsimonious, and easy to understand. Parsimony ensures that overlapping dimensions are removed and only those dimensions which provide explanatory and discriminatory power are considered.
- e) **Design assistance** - It is necessary for the framework to identify broad design parameters and guidelines for design for the various categories.
- f) **Utility** - The utility of the framework may be determined by analyzing how efficiently the present systems are mapped on to the framework and how effective it is in

identifying research issues and potential areas for research.

The proposed framework is evaluated below on the various criteria described above.

Acceptance: The three dimensions represented along the three axes of the framework are well established components of a knowledge based management system (Brodie and Mylopoulos, 1986). The classifications of each of the dimensions are based on accepted categories in research literature (Brodie and Mylopoulos, 1986; Date, 1990; Stonebraker and Hearst, 1988). Thus we can claim that there is sufficient scientific consensus on these classifications. Also, these classifications are well understood by practitioners, since distinct commercial systems are available in each of these categories independently.

Precision: The three dimensions are to a large extent unambiguous and orthogonal to each other. Obviously not all cells are equally attractive and the current state of research has tended to use certain combinations of the three subdimensions. For example, rule-based and logic representation tends to be linked to relational data model due to sharing of common logic formalisms. However, there is now an increasing tendency to develop hybrid structures with multiple representations in KBMS. For instance, using rule-based representation inside an object oriented or frame-based representation scheme as in HIPAC (McCarthy & Dayal, 1988).

Generality: The classification under each of the dimensions is quite exhaustive and most systems developed so far would be able to fit into one of the cells. The mapping of the operational systems onto the framework establishes to some extent the validity of the framework. However, since the area is rapidly evolving and new knowledge representation schemes and data models are being continuously explored, new categories may have to be added to each of the dimensions. Also, there is a tendency now to use multiple knowledge representation schemes, in very complex systems, to address different subproblems in the same system. If we consider the total system as one system, it is very difficult to map it onto the framework. However, they may be considered as separate sub-systems and hence may be individually mapped. Indeed, the design issues and guidelines are different for each of the subsystem and need to be addressed at the sub-system level only.

Parsimony: The dimensions of the framework have to be parsimonious with minimum overlap between dimensions. This is required to facilitate mapping of research systems into the different cells. The three dimensions in the proposed framework have very little overlap, since they represent three different components of the system, and therefore can be considered to be parsimonious in their definition.

Design Assistance: The framework provides broad guidelines for design, as it contains the three major components of a knowledge based management system. We have highlighted the characteristics of the problems that are addressed by different integration architectures. Design guidelines for different categories of problems have been proposed while describing the features of the various data models and knowledge representation schemes. Also, during the discussion on the individual cells of the framework we have highlighted the significant features of the systems developed.

The framework provides initial guidance in selecting a particular design option given the characteristics of the problem and various other constraints and considerations of a particular problem context. For example, if an organization is interested in developing a KBMS based on data presently stored in a relational database, then using the framework we can decide that the most appropriate knowledge representation scheme could be rule-based or logic. Also, since the data is part of the corporate database and is used for other applications in the organization, a loosely coupled architecture may be more appropriate in that context.

Utility: The use of the framework is dependent on how efficiently it is able to map the system developed so far on to the framework. The mapping of the existing systems, as described in the last section, shows that these systems can be efficiently classified along these dimensions and the framework provides some explanatory power in describing the characteristics of these systems and identifying research issues and solutions based on past experience. The utility of the framework in terms of identifying issues for future research is discussed in the next section.

Implications for Research and Practice

The framework is useful in identifying potential areas for further research. An analysis of the framework clearly reveals that most of the research thrust is on relational database systems with some interest in object oriented systems. It is generally believed that the future of KBMS is dependent on the integration of KBS with DBMS. KBMS to make major in-roads into traditional applications have to be integrated with available DBMS in organizations such as a relational DBMS or the more prevalent network or hierarchical DBMS. A significant number of the older DBMS are based on either hierarchical or network data model and developing interfaces to integrate these with KBS will provide significant growth in the use of KBMS. Application research focused on integrating KBS with these data models should result in greater success in the adoption of expert systems in commercial applications. This alternative short-term strategy may be useful for the industry, especially in

light of the criticisms leveled on expert systems inability to provide significant benefits to organization, thereby resulting in very slow growth in demand for these systems.

Research in KBMS has primarily concentrated on integrating KBS with only relational data models. A major reason for this could be that its data model is more amenable to mathematical formalisms compared to other models. Further, the growing popularity of relational data models in commercial DBMS may have also provided the impetus to study relational DBMS. The ability to link relational databases with object-oriented knowledge representation schemes would provide a good fit between an extensively available commercial DBMS and a representation scheme that is rich, with excellent features to exploit the potential of AI technology.

In recent times, there is increased research emphasis on object oriented data models. Although research in the development of object oriented systems may develop very efficient and versatile systems, its markets will be limited due to its portability problems to existing DP environment. These systems will be effective as stand-alone systems, but will have problems in integrating with existing DBMS in the organization. Hence, although beneficial in the long run it may not be a commercial success in the short run.

For the practitioner, the framework provides a broad perspective of the state of the art in this field and identifies the various options available to the user. Given a certain context for KBMS development, the user can evaluate the context and determine: (a) the appropriate knowledge representation scheme that could easily model the problem, (b) the database (if any) that need to be accessed and the data model characteristics of the database, (c) the type of information access requirements and complexity of the interface between DBMS and KBS to evaluate the coupling option, and (d) to design an overall system architecture based on the decisions made in the previous steps.

Conclusion

KBMS has gained importance in the last few years, primarily from the enhanced capabilities achieved through integration of the two technologies - artificial intelligence and DBMS. This paper developed an integrated framework that links the knowledge based systems with the database models and the architectures used for linking the two systems. Various research systems were mapped onto the framework. The mapping revealed that some research areas are more popular than others. It identified areas for future research and also areas where commercial exploitation would result in rapid acceptance of KBMS.

References

Bachant, J. & McDermott, J. (1984). R1 revisited: Four years in the trenches. *The AI Magazine*, 5, 21-32.

- Barker, V.E., & O'Connor, D.E. (1989). Expert Systems for configuration at digital: XCON and beyond. *Communications of the ACM*, 32(3), 298-318.
- Bates, M., M.G. Moser, & D.Stallard. (1986). The IRUS transportable natural language database interface. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- BIM (1986). BIM Prolog Manual. BIM, Belgium.
- Bobrow, D.G., & Winograd, T. (1977). An Overview of KRL, A Knowledge Representation Language. *Cognitive Science*, 1(1), 3-46.
- Brodie, M.L. & J. Mylopoulos (1986). *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*, Springer-Verlag, New York.
- Brodie, M.L. & J. Mylopoulos (1986b). Knowledge Bases and Databases: Semantic vs. Computational Theories of information. In *New Directions for Database Systems*, (ed.) Ariav G. & J. Clifford, Ablex Publishing Corporation, Norwood, NJ. 186-218.
- Brodie M.L. (1986c). Knowledge based management systems - Panel discussion. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Bocca, J. (1986). EDUCE - A Marriage of Convenience: Prolog and a Relational DBMS. In *Proceedings of the 1986 Symposium on Logic Programming*, Salt Lake City, Utah. 36-45.
- Carey M.J., D.J. Dewitt, & G. Graefe. (1986). Mechanism for concurrency control and recovery in Prolog. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Ceri S., Gottlob G., & Wiederhold G. (1986). Interfacing Relational Databases & Prolog efficiently. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Chakravarthy U.S., D.H. Fishman, & J. Minker. (1986). Semantic query optimization in expert systems and database systems. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Chang C.L. & A. Walker. (1986). PROSQL: A prolog programming interface with SQL/DS. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Chobineh, J., & Sen, A. (1987). A framework for deductive database design in decision support systems. *Proceeding of the Eighth International Conference of Information Systems*, Pittsburgh, PA. 241-251.
- Codd, E.F. (1979). Extending the database relational model to capture more meaning. *ACM Transaction of Database Systems*, Vol. 4, No. 4, 395-434.
- Copeland G., & Maier, D.. (1984). Making Smalltalk a Database System. *ACM SIGMOD*. 316-325.
- Dahl, V. (1982). Database System Development through Logic. *ACM Transaction on Database Systems*, 7, 102-123.
- Date, C.J. (1990), *An Introduction to Database Systems*, 4th edition, Addison Wesley, Reading, MA.
- Davis R. (1980). Content referencing: Reasoning about rules. *Artificial Intelligence*, 15(3), 223-239.
- Dayal U., & J.M., Smith (1986). PROBE: A Knowledge-Oriented Database Management System. In *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*, (ed.) Brodie, M.L. & J. Mylopoulos, Springer-Verlag, New York.
- Fikes, R., & Kehler, T., (1985). The Role of Frame-Based Representation in Reasoning. *Communication of the ACM*, 28(9), 904-920.
- Fishman, D.H. (1986). The DBMS-Expert System Connection. *New Directions for Database Systems*, (ed.) Ariav G. & J. Clifford, Ablex Publishing Corporation, Norwood, NJ.
- Fox M.S. & J. McDermott. (1986). The role of databases in knowledge based systems. In *On knowledge base management systems: Integrating artificial intelligence and data base technologies*, (ed.) Brodie M.L. & J. Mylopoulos, Springer-Verlag, New York.
- Fox, M.S. (1986). Experiences with SRL: An Analysis of Frame-Based Knowledge Representation. *Expert Database Systems* (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Friedland P. (1985). Special Edition on Architectures for Knowledge Based Systems. *Communications of the ACM*, 28(9).
- Gallaire, H., J. Minker, & J. Nicolas (1987). (eds.), "Logic and Databases: A Deductive Approach," *ACM Computing Surveys*, 16(2).
- Genesereth, M.R., Ginsberg, M.L.. (1985). Logic Programming. *Communications of the ACM*, 28(9), 933-941.
- Goldstein I.P., & Bobrow D.G. (1980). Descriptions for a programming environment. *Proceedings of AAAI-80*, Stanford, CA.
- Gorry A., & Scott Morton. (1971). A Framework for Management Information Systems. *Sloan Management Review*. Fall, 55-70.
- Grant J., & J. Minker. (1981). Optimization in deductive and conventional database systems. *Advances in Database Theory*, eds. Gallaire H., J. Minker, & J.M. Nicolos, Plenum Press, New York.
- Grishman, R. (1978). The simplification of retrieval requests generated by question answering systems. *Proceedings of 4th VLDB conference*. 400-406.
- Hammer, M.M., & McLeod D.J. (1981). Database Description with SDM: A Semantic Database Model. *ACM Transactions on Database Systems*, 6(3).
- Hayes-Roth, F. (1985). Rule-Based Systems. *Communications of the ACM*, 28(9), 921-932.
- Ioannidis, Y.E., & J. Chen, M.A. Friedman, Tsangaris, M.M. (1988). BERMUDA - An Architectural Perspective on Interfacing PROLOG to a Database Machine. in *Expert Database Systems: Proceedings from the Second International Conference*, (ed.) Kerschberg, L. Benjamin/Cummings, Menlo Park, California. 229-236.
- Ives B., Hamilton S., Davis G.B. (1980). A Framework for Research in Computer- Based Management Information Systems. *Management Science*. 26(9). 910-134.
- Jarke M. (1986). External semantic query simplification. *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Jarke, M., & Y. Vassiliou. (1986). Databases and Expert Systems: Opportunities and Architectures for Integration. *New Application of Databases*, Academic Press, London.
- Jarke M., J. Clifford, Y. Vassiliou. (1984). An optimizing front end to a relational query system. *ACM SIGMOD*.

- Jarke, M., J. Koch, M. Mall, J.W. Schimdt. (1982). Query optimization research in the database programming languages project. *Database engineering*, 5(3), 11-14.
- Kehler, T.P., & Clemenson, G.D. (1984). An application development system for expert systems. *Systems Software*, 3(1), 212-224.
- Kellog, C. (1982). Knowledge Management: A practical amalgam of knowledge and database technology. *Proceedings of AAAI-82*, Pittsburgh.
- Kerschberg, L. (1988). *Expert Database Systems: Proceedings from the Second International Conference*, Benjamin/Cummings, Menlo Park, CA.
- Kerschberg, L. (1986). *Expert Database Systems: Proceedings from the First International Workshop*, Benjamin/Cummings, Menlo Park, CA.
- King R. (1986). A database management system based on an object-oriented model. *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- King, R., & McLeod, D. (1985). Semantic Data models. *Database Design*, (ed.) Yao. Prentice Hall, New York.
- Kung R. (1986). Heuristic search in database systems. *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Kunifujii, S. & H. Yakota. (1982). Prolog and relational databases for fifth generation computer systems. *Proceedings of the workshop on logical bases for databases*.
- Lafue, G.M.E. (1983). Basic decisions about linking an expert system with a DBMS: A case study. *Database engineering*, 6(4).
- Lafue G.M.E., & R.G. Smith. (1986). Implementation of a semantic integrity manager with a knowledge representation system. *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Lerch J.F., & Mantei M.M. (1984). A Framework for Computer Support in Managerial Decision Making. *Proceedings of the Fifth International Conference on Information Systems*, 129-139.
- Quintus (1987). *Quintus Prolog Database Interface Manual*. Quintus Computer Systems, Mountain View, CA.
- Mason R.O., & Mitroff, I.I. (1973). A Program for Research on Management Information Systems. *Management Science*. 19(5), 475-487.
- McCarthy, D., & Dayal, U. (1989). "The Architecture of an Active Database Management System," *Proceedings of SIGMOD conference*, Portland, May, 215-224.
- Minsky M. (1975). A Framework for Representing Knowledge. In *Psychology of Computer Vision*, (ed.) Winston P.H. McGraw-Hill, New York.
- Moser M. (1984). Domain dependent semantic acquisition. *Proceedings of first conference on AI applications, IEEE.*, 1984.
- Mylopoulos J. (1984). An overview of knowledge representation. *On conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*, (ed.) Brodie M.L., Springer-Verlag, New York.
- Naqvi S.A. (1986). Negation in knowledge base management systems. *On knowledge base management systems: Integrating artificial intelligence and data base technologies*, (ed.) Brodie M.L. & J. Mylopoulos, Springer-Verlag, New York.
- Napheys, B., & Herkimer, D. (1988). A Look at Loosely-Coupled Prolog Database Systems. In *Expert Database Systems: Proceedings from the Second International Conference*, (ed.) Kerschberg, L. Benjamin/Cummings, Menlo Park, California. 259-271.
- Nicolos, J.M., & Yazdanian, N. (1978). Integrity checking in deductive databases. *Logic and Databases*, (eds.) Gallaire H. & J. Minker, Plenum, New York.
- Olson, J.P., S.P. Ellis. (1982). Probbwell, An expert advisor for determining problems with producing wells. *Proceedings IBM scientific/engineering conference*, New York, 95-101.
- Parasaye, K. (1983). Database management, knowledge base management and expert system development in Prolog. *ACM SIGMOD*. 159-178.
- Parasaye, K., Chignell, M., Khoshafian, S., & Wong, H. (1989). *Intelligent Databases: Object-Oriented, Deductive, Hypermedia Technologies*. John Wiley.
- Peckham J. & F. Maryanski. (1988). Semantic Data Models. *ACM Computing Surveys*. 20(3), 183-190.
- Prerau D.S. (1985). COMPASS: An expert system for telephone switch maintenance. *Expert Systems: The International Journal of Knowledge Engineering*. 2(3). 112-126.
- Raschid, L. & S.Y.W. Su. (1988). "A Transaction Oriented Mechanism to Control Processing in a Knowledge Base Management System," In *Expert Database Systems: Proceedings from the Second International Conference*, (ed.) Kerschberg, L. Benjamin/Cummings, Menlo Park, California. 353-373.
- Reiter, R. (1978). Deductive question-answering on relational databases. *Logic and Databases*, (eds.) Gallaire H. & J. Minker, Plenum Press, New York.
- Rowe, N.C. (1983). Top-down Statistical Estimation on a Database. In *Proceedings of 83 SIGMOD*. Baltimore, MD. 135-145.
- Sciore E. & D.S. Warren. (1986). Toward an integrated database prolog system. In *Expert Database Systems*, (ed.) Kerschberg, L., Benjamin/Cummings Publishing Co. Inc., Menlo Park, CA.
- Shepherd, A., Kerschberg, L. (1984) Prism: A Knowledge Based System for Semantic Integrity Specification and Enforcement in Database Systems. *ACM SIGMOD*. 307-315.
- Shipman D. (1981). The Functional Data Model and the Data Language DAPLEX. *ACM Transactions On Database Systems*, 6(1).
- Spark-Jones K & B. Boguraev. (1982). A natural Language analyser for database access. *Information and Technology: Research and Development*, Vol. 1.
- Steinberg, D. (1990) AMEX builds an expert system to assist its credit analysis. in *Management Information Systems - Reading and Cases - A Managerial Perspective* (eds.) Boynton, A. & Zmud, R.W. Scott, Foresman & Co., Glenview, Illinois. 139-143.
- Stonebraker, M. (1982). A Rules System for a Relational Database System. In *Proceedings of Second International Conference on Databases*. Jerusalem.
- Stonebraker, M., & Hearst, M. (1988). Future Trends in Expert Database Systems, In *Expert Database Systems: Proceedings from the Second International Conference*, (ed.) Kerschberg, L. Benjamin/Cummings,

Menlo Park, California. 3-20.

Stonebraker M. (1986). Triggers and inferences in database systems. *On Knowledge based management system: Integrating artificial intelligence and data base technologies*, Brodie M.L., & J. Mylopoulos, Springer-Verlag, New York.

Tsur, S., Zaniolo C. (1984). An Implementation of GEM - Supporting a Semantic Model on a Relational Back-end. *ACM SIGMOD*. 286-295.

Ullman J.D. (1986). An approach to processing queries in a logic based query languages. *On knowledge base management systems: Integrating artificial intelligence and data base technologies*, Brodie M.L., & J. Mylopoulos, Springer-Verlag, New York.

Vassiliou Y. (1986). Knowledge based and database systems. *On knowledge base management systems: Integrating artificial intelligence and data base technologies*, (eds.) Brodie, M.L., & J. Mylopoulos, Springer-Verlag, New York.

Vesonder G.T., Stolfo S.J., Zielinski J.E., Miller F.D., & Copp D.H. (1983). ACE: An Expert System for Telephone Cable Maintenance, *Proceedings of*

the Eight Joint Conference on Artificial Intelligence. 116-121.

Warren, D. (1986). Panel discussion. *On knowledge base management systems: Integrating artificial intelligence and data base technologies*, (eds.) Brodie, M.L., & J. Mylopoulos, Springer-Verlag, New York.

Widerhold, G., & El-Masri, R. (1979). Structural Model for Database Design. *Proceedings of International Conference on the Entity-Relationship Approach to System Analysis and Design*, Los Angeles.

Woods W.A. (1986). Knowledge base retrieval. *On knowledge base management systems: Integrating artificial intelligence and data base technologies*, Brodie, M.L., & J. Mylopoulos, Springer-Verlag, New York.

Zaniolo, C. (1986). Object Oriented Database Systems and Knowledge Systems. In *Expert Database Systems* (ed.) Kerschberg L., Benjamin/Cummings Publishing Co. Menlo Park, CA.

The author gratefully acknowledges the input provided by Govinda Rao in the initial stages of the project.

G. Premkumar is an Assistant Professor in the College of Business Administration at Iowa State University. He received his Ph.D. in MIS from the Katz Graduate School of Business at the University of Pittsburgh. He has a graduate degree in management and an undergraduate degree in engineering from India. Dr. Premkumar has published papers in Decision Sciences, Information and Management, Omega - International Journal of Management Science, Long Range Planning, and International Journal of Man Machine Studies. He has also presented papers in national and international conferences. His current research interests include decision support systems, knowledge base systems, and IS planning.

Related Content

A Graph-Based Approach for Semantic Process Model Discovery

Ahmed Gater, Daniela Grigori and Mokrane Bouzeghoub (2012). *Graph Data Management: Techniques and Applications* (pp. 438-462).

www.irma-international.org/chapter/graph-based-approach-semantic-process/58622/

A Study of a Generic Schema for Management of Multidatabase Systems

Shirley A. Becker, Rick Gibson and Nancy L. Leist (1996). *Journal of Database Management* (pp. 14-20).

www.irma-international.org/article/study-generic-schema-management-multidatabase/51169/

How NoSQL Databases Work

(2018). *Bridging Relational and NoSQL Databases* (pp. 124-175).

www.irma-international.org/chapter/how-nosql-databases-work/191982/

A Rigorous Approach for Mapping Workflows to Object-Oriented IS Models

Staf Joosten and Sandeep Purao (2002). *Journal of Database Management* (pp. 1-19).

www.irma-international.org/article/rigorous-approach-mapping-workflows-object/3284/

Web Services, Service-Oriented Computing, and Service-Oriented Architecture: Separating Hype from Reality

John Erickson and Keng Siau (2010). *Principle Advancements in Database Management Technologies: New Applications and Frameworks* (pp. 176-188).

www.irma-international.org/chapter/web-services-service-oriented-computing/39355/