An Object-Oriented Design for Distributed Knowledge-Based Systems

David A. Carlson  Sudha Ram

Department of Management Information Systems
College of Business and Public Administration
University of Arizona
Tucson, AZ 85721
(602)621-2748

Abstract
Many management information systems have requirements that are inherently distributed, either logically or physically, however, no adequate architectures exist which allow multiple knowledge bases and databases to be shared and integrated. An architecture for Distributed Knowledge-Based Systems (DKBS) is presented which satisfies these requirements. A Distributed Knowledge Base Management System (DKBMS) is described which manages the meta-knowledge necessary to coordinate multiple local Knowledge Based Systems (KBSs). We propose an object-oriented design for implementing the DKBMS structure and the communication protocol which connects the DKBMS with each of the local KBSs. An example is described where the DKBMS architecture is applied to planning research projects within a university. The resulting architecture is compared with previous research in related topics, and directions for future research on the DKBS architecture are described.

1 Introduction
This paper presents an architectural design which facilitates the implementation of information systems supporting knowledge-based decisions made by groups of individuals or business entities. This architecture allows both knowledge-based and data-based information to be shared and integrated by semi-autonomous sites in order to achieve common global objectives.

A Distributed Knowledge-Based System (DKBS) consists of several Knowledge-Based Systems (KBSs) that may be logically or physically distinct from one another. The KBSs are linked together and managed by a single Distributed Knowledge Base Management System (DKBMS) which coordinates global inference to resolve goals that span multiple KBS sites in the network. A significant requirement of this global inference process will be to resolve the inevitable conflict which arises between the autonomous KBSs. A global knowledge base component of the DKBMS will contain meta-knowledge describing the entire distributed network of knowledge-based systems. User goals may be directed at either the DKBMS or at a specific local KBS. We propose that an object-oriented model be used to implement the DKBMS structure.

We present an application for a distributed knowledge-based system which assists research personnel in planning research projects and their funding. Although our DKBS architecture supports any knowledge representation paradigm at each local KBS, we will present an object-oriented design for the research planning system. Object-oriented programming (OOP) supplies a powerful control mechanism for organizing the KBS; rule-based and logic-based representations may be used to store the knowledge within the system.

Section 2 describes our motivation for research on Distributed Knowledge-Based Systems and reviews previous research in related areas. Section 3 presents a conceptual design of the DKBMS architecture and details of the DKBMS specification. Section 4 describes a knowledge-based system for planning research projects and proposes an object-oriented approach for implementing the prototype design. Finally, we discuss open research issues and present conclusions.

2 Motivation for Research in DKBS
This section describes our research goals, provides a brief overview of knowledge and data integration, and reviews prior research which is relevant to the architecture.

2.1 Research Goals
Our goal is to design an architecture which facilitates the implementation of information systems supporting decisions made by groups of individuals or business entities. This group decision process requires a network of databases and knowledge bases with coordinated communication among them. Cooperative action needs to be stressed. Performance of the integrated information system may be enhanced by allowing parallel processing of sub-tasks by individual knowledge-based systems. When cooperative decisions are required, each knowledge-based system must have access to multiple external sources of data and knowledge. No adequate architectures exist in the current literature to integrate multiple distributed knowledge base and database systems.

The knowledge-based component of this integrated system would contain meta-knowledge describing decision heuristics and empirically established relationships between data items. These relationships would infer information from conflicting or partial data, or manage uncertainty associated with the stored data. Information management problems require procedural knowledge for determining acceptable solutions, and control knowledge in terms of strategies for making efficient use of the procedural knowledge. The meta-knowledge would direct the global inferences and resolve conflict among multiple autonomous knowledge-based systems.

In order to establish an effective communication between knowledge base and database systems, each component must depend on its own representation structures and inference methods. A simple interface between the two components will not suffice. In an integrated system the strengths of each system must be exploited [32].
The authors state that "knowledge sharing" will be new to both controlling and general information, typically complex and relatively small in volume, whereas, data represent the manipulated and factual information, typically regular but voluminous [33]. The KBMS will require unified control schemes for consistency, semantics, and knowledge content (i.e., what knowledge resources the KBMS has) as well as for redundancy, reliability, and security. Several requirements are outlined which will allow a KBMS to manage the "knowledge resources" of a collection of knowledge based applications. The solutions to a series of objectives for KBMS architectures are left for future research.

2.2 Previous Research

Our architectural design represents a synthesis of four major areas of research: distributed database systems, distributed problem solving, object-oriented programming, and strategic planning.

Distributed Database Systems. Many issues discussed in the Distributed Database Systems (DDBS) literature are also relevant to DKBS, namely, the global schema definition, communication among local databases, heterogeneity, fragmentation, and allocation. The three-level architecture described by Ceri and Pelagatti [3] defines a conceptual schema which contains relations and attributes for the global network of distributed relational databases. The second and third levels of Ceri's architecture define fragmentation and allocation schemas which distribute the conceptual schema to the various sites in the network. All of these schemas are represented using consistent global semantics. Finally, the site schemas are mapped to the data definition language used by the DBMS at each particular site. Several architectures have been proposed for distributed database systems [23, 20], the most accepted of which is the three-level architecture described by Ceri, et al [3].

Distributed Problem Solving. Distributed problem solving requires an architecture for a network of independent knowledge-based systems that may cooperate to solve a problem faced by one individual node or a subset of the nodes. A survey of research in this field is described by Gasser [12], Smith [24], and Sritharan [25], and specific research topics can be found in [4, 5, 6, 7, 8, 10, 11, 15, 18, 19].

Partial global plans are discussed by Durfee and Lesser [7, 8] as a method for coordinating distributed problem solvers. A network plan is described as containing a local plan representing the detailed plan maintained by a node that is pursuing the plan, a node-plan representing the less specific plan that nodes use to communicate their intentions, and a partial global plan that represents how several nodes are actually working toward a larger goal. The partial global planner scans models of other nodes to find cases where several nodes are working on parts of some larger partial global goal. The partial global planner forms partial global plans to achieve the partial global goals. A partial global goal is global in the sense that it may (but does not necessarily) encompass the local goals of several nodes, and is partial in that only part of the network might participate in it.

Object-Oriented Programming. Objects are entities that combine the properties of procedures and data since they perform computations and save local state. All activity in object-oriented programming (OOP) results from sending messages between objects: a form of indirect procedure call. A set of class messages is designed to provide the complete user interface with an object. This related set of messages (or operators) forms a protocol for the object. The term polymorphism is used in object-oriented programming to refer to the capability for different classes of objects to respond to exactly the same protocols [26, 14, 31, 29].

The object-oriented paradigm appears to be an appropriate choice for designing a global knowledge base which will integrate a variety of distributed, heterogeneous knowledge bases. The object-oriented approach does not favor any particular "knowledge paradigm" such as logic, frames, semantic nets, or productions systems. Therefore, the object-oriented approach may be used to provide not only external views, but also internal representation of the knowledge base so that specialized interpreters can be used for reasoning about the knowledge [34].

DSS for Strategic Planning. Various approaches have been described for developing information systems to support the strategic planning domain. Goul, et al [13] describes a knowledge-based Decision Support System (DSS) that guides, instructs, and provides reasoning to identify strategic opportunities, problems, and crises. The authors examined the effects of using a knowledge-based DSS on performance in the "problem-finding" or initial stage of the strategic planning process. Applegate, et al [1] describes the requirements for knowledge management in organizational planning, and presents a knowledge-based planning system that has been implemented within a Group Decision Support Systems (GDSS) environment. McIntyre, et al [22] describes knowledge-based techniques using Semantic Inheritance Nets for view integration and for providing a flexible, dynamic, automated model of the Information Systems (IS) planning environment. A frame-based system is described by Carlson and McIntyre [2] to enhance modeling and inferencing in certain strategic planning tasks, and a strategic planning process is described for identifying organizational needs in order to increase the organization’s effectiveness.

3 Architecture for Distributed Knowledge-Based Systems

This section describes the architectural design for the overall Distributed Knowledge-Based System and for the Distributed Knowledge Base Management System which integrates the network of independent Knowledge-Based Systems.

3.1 A Distributed Knowledge-Based System

A Distributed Knowledge-Based System consists of several Knowledge-Based Systems that may be logically or physically distinct from one another. The Knowledge-Based Systems (KBS) are integrated and managed by a single Distributed Knowledge Base Management System (DKBMS), as depicted in Figure 1. Each KBS consists of an inference engine, one or more knowledge base(s), database(s), and user interface(s). A Knowledge Base Management System (KBMS) provides a mechanism and a syntax to define and access the knowledge bases and databases. The databases may also be accessed directly by interfacing with a Database Management System (DBMS), or with a Distributed Database Management System (DDBMS). Each
KBS may be implemented using a different KBMS. Thus, the DKBMS must be capable of communicating in a heterogeneous system where queries and inference goals are translated between the representational languages used by the DKBMS and by each local KBMS.

Users may interact with the DKBMS or directly with any one of the local KBMS. In a direct query to a local KBS, if the local inferences require the use of other KBs, communication must take place between the local KBMS and the DKBMS. The DKBMS passes messages to the relevant KBs(s), consolidates the results, and passes results back to the requesting KBMS. Inferences for queries issued directly to the DKBMS are decomposed into inferences at one or more local KBs, the results of which must be returned to the DKBMS for integration and formulation of a user response. Thus, the inferencing process can take advantage of parallelism, while minimizing communication between local and global levels of the architecture.

### 3.2 A Distributed Knowledge Base Management System

The DKBMS is composed of three components: the Global Knowledge Base (GKB), the Knowledge Cluster Dictionary, and the Communication Protocol (see Figure 2). The global knowledge base contains knowledge describing the overall system organization and meta-knowledge defining the scope of the knowledge managed by each individual KB. The meta-knowledge would also resolve conflicts between local KBs through heuristics and through authority relationships established within the KB organization. User queries may be directed at either the DKBMS, or at a specific local KB. When a user submits a goal to the DKBMS, the global knowledge base will use the Knowledge Cluster Dictionary to determine which, if any, local knowledge bases contain the knowledge required to address the goal that was stated in the query. Several local KBs may be solicited by the GKB in order to complete a single query. The GKB should contain the meta-knowledge necessary to select the local KBs which are most likely to resolve the user’s goal. Absolute delineation of a knowledge base’s capabilities is still an active field of research. Thus, the GKB must be designed to maximize resolution of its goals by local knowledge bases through a careful design of the GKB meta-knowledge.

External views of the GKB may be defined to facilitate or to control user queries. The actual external view definition will be determined by the particular knowledge representation paradigm selected for implementing the GKB. Our prototype system uses frames to segment the inference processes within the GKB. An external view may be limited to one or more of these knowledge frames. Defining several external views would allow knowledge to be segmented for different categories of users, e.g., confidentiality of specific goals for a business plan.

The second tier of the architecture defines a Knowledge Cluster Dictionary which groups the GKB subgoals into logically related clusters, which we will refer to as knowledge clusters. The GKB subgoals are the most detailed goals resolved from the initial user goal and the GKB meta-knowledge. The knowledge clusters are defined as part of the design of a particular DKBMS application. The Knowledge Cluster Dictionary is composed of two parts: assignment of the GKB subgoals to a particular knowledge cluster, and assignment of each knowledge cluster to one or more local KBs. By assigning a knowledge cluster to a local KB, the cluster subgoals are also implicitly assigned to that KB. Therefore, the KB is expected to contain the knowledge needed to resolve those subgoals. The architecture accommodates redundant knowledge among the individual sites by allocating a knowledge cluster to multiple sites.

The third tier of this DKBMS architecture consists of the communication link between the DKBMS and the local KBs. The local KBs will communicate with the DKBMS via a standard message protocol, and the local sites will use the DKBMS as a liaison to either instruct or query other local KBs. Depending on the message sent by the initiating local site, a specific node may be requested as the message target, or the GKB may use its organizational knowledge to select zero, one, or many sites as targets. Zero sites would be selected when the GKB meta-knowledge is unable to identify a local KB capable of resolving the original goal.

Heterogeneous knowledge representations will be supported at
the local KBSs by providing a common communication protocol and argument mapping function between the GKB representation and each local KBS representation. Inherent in the communication protocol is a mapping algorithm which translates message terms between languages. For example, if the local KBS is located in Japan and the DKBMS is located in the U.S.A., then Yen must be translated to US Dollars, meters to feet, etc.

A significant benefit provided by this three tier architecture is that a logical separation is achieved between the DKBMS and the local KBSs. If the KBSs are added or deleted from the network, subdivided, or extended within the original application domain, then only the Knowledge Cluster Dictionary must be modified; the GKB is unaffected. Conversely, if the GKB meta-knowledge is reorganized, then the subgoals may need to be reassigned to knowledge clusters in the Knowledge Cluster Dictionary, but the KBSs are isolated from the changes.

The conceptual design of this architecture may be implemented with any knowledge representation paradigm; the actual representation selected for a particular application should be determined by the specific information requirements. For example, the Knowledge Cluster Dictionary may be implemented by either an object-oriented data structure, or by an external relational database. The next section discusses the design proposed for our prototype system.

4 A DKBS for Planning Research Projects

We now present an application domain which would benefit from a distributed knowledge-based system, and propose an object-oriented prototype design.

4.1 A Research Planning Domain

Most research personnel, especially those in a managerial role, devote considerable effort to managing a portfolio of research projects. Of these management tasks, planning future research projects and coordinating these projects with other organizations presents the most difficult information management problem. Most plans, whether for research or strategic business planning, must integrate the resources and requirements of numerous organizational units. The individual plans of these autonomous units must be linked between hierarchical levels, across departments and functional areas, and among outside organizations.

Consider a DKBS to support research planning at a university. This environment might have four local KBSs: Management Information Systems department, Electrical Engineering department, a government funding agency, and a private corporation (see Figure 3). An administrative Research Planning department has the charter to plan and coordinate university-wide research. This research planning department does not maintain its own KBS, however, it accesses the four local KBSs through the DKBMS interface.

Typical planning tasks would include:

- Establishing goals for the direction of research projects
- Establishing goals for total research funding in an organization
- Reviewing progress toward established goals
- Searching for sources of funding for new or continuing projects
- Submitting funding proposals
- Coordinating several independent, but related, projects
Goal 1: Develop a research plan for the next five years.
Goal 2: Plan new faculty recruiting efforts to match research objectives.
Goal 3: Plan computer facility requirements needed to support research plans.

Table 1: Goals resolvable by an individual KBS

Goal 1: Establish a joint center for collaborative research with industry.
Goal 2: Prepare a research grant proposal for a government agency to support a multi-department project.
Goal 3: Establish a task force to design a campus-wide computer network.

Table 2: Goals requiring input from multiple KBSs
- Investigating prior and current research in related areas
- Reviewing the progress on research projects

This planning problem provides an interesting domain for showing the feasibility of our research, in several respects. First, the problem is inherently distributed; the planning entities are logically and geographically dispersed. Second, the planning agents are semi-autonomous. Third, atomic data are not an adequate representation for modeling the interaction of the planning agents, thus knowledge-based paradigms are required. Fourth, the representational requirements of the individual agents are not uniform. The agents vary in size, complexity, deterministic characteristics, and stability, hence, heterogeneity is desirable at the site-level of the model. Finally, coordination and coherence of the global system is desirable. Therefore, these distributed, semi-autonomous, intelligent, heterogeneous planning entities must cooperate to produce a coherent plan and to resolve conflict.

The DKBS may be faced with two categories of goals: those which can be resolved by any one local KBS without input from outside sources, and those goals which require either knowledge-based and/or data-based input from other KBSs. Table 1 displays a set of possible goals falling into the first category and Table 2 displays several goals requiring input from multiple KBSs. The goals in Table 2 may not be resolved by a single KBS because it has insufficient knowledge resources, there exist potential conflicts with other KBSs, or both. These two categories of goals will be addressed again in our discussion of DKBMS and KBS integration in section 4.4.

We present an object-oriented design for modeling the above planning requirements and describe the knowledge-based support that such a model might contain.

4.2 An Object-Oriented Design for a Local KBS
This model will not attempt to represent all features that are required to effectively support research planning activities. Rather, this model describes the primary classes in the object-oriented design in order to demonstrate the feasibility and value of the DKBS architecture.

The model design specifications must be formalized in terms of an object-oriented classification structure. The object-oriented paradigm consists of two primary components: message-passing and specialization. An object is defined as an instance of a class, and the class itself may be a subclass within a larger classification hierarchy. Each class defines its own instance variables and methods (a method is essentially a procedure which is activated by a message sent to an object), and both variables and methods may be inherited from higher levels of the class hierarchy. An object is a specific instance of a class, and the object encapsulates all variables and methods defined (local or inherited) for that class.

The primary classes in our prototype KBS are shown in Figure 4. Each planning entity is represented in one instance of the Entity class. This object stores pointers to all other objects required to represent the activities and plans of the single entity. An instance of the Person object is created for every member of the entity staff, and these pointers to Person objects are stored as a collection in the 'staff' instance variable of the Entity object. Figure 5 displays these object relationships, among others.

Figure 4 shows a Person object being referenced by two other objects: Entity and Project. Since the Person object encapsulates all relevant knowledge about a single person, any other object which references Person will also have access to that Person object's characteristics through its methods. These other objects would access Person's methods by sending a message to the Person object.

The Person object maintains pointers to two other classes of objects: Assertion and Rule. These objects are used to model the opinions and heuristics held by a particular individual, and these additional objects may also be available for inspection by Entity and Project to determine the knowledge-based behavior of that person. Thus, the object-oriented representation effectively models the entity's components and the knowledge-based behavior of selected components.

Although we described an object-oriented representation for the local KBS in this paper, our proposed DKBS architecture supports any appropriate knowledge base or database implementation. This knowledge representation does not attempt to create a system which produces "The Plan," but rather, the knowledge base will store rules and relationships which assist the plan developer in evaluating his plan's coherence with all relevant plans in the distributed system.

4.3 An Object-Oriented DKBMS Design
An object-oriented design approach was selected for our prototype DKBMS implementation. Object-oriented programming (OOP) provides an intuitively appealing environment for modeling the objects and the communication between objects in a distributed system. OOP provides a natural approach for frame-based representation in the GKB design, supports heterogeneous knowledge rep-
representations in the local knowledge bases, and facilitates a message passing paradigm for the communication protocol which connects the components of the DKBMS architecture.

The primary classes for our DKBMS design are shown in Figure 6. Single instances of the GKB and the DKBMS classes are created to manage all objects associated with the DKBMS architecture. Similarly, single instances of the ClusterDict and LanguageDict classes are created to manage the Knowledge Cluster Dictionary and the Language Mapping Dictionary, respectively. The Frame objects are created as required to define the meta-knowledge within the GKB; the Slot class is instantiated to define the slots within each Frame object. The Cluster class describes the contents of each knowledge cluster, which is used to group subgoals within the Knowledge Cluster Dictionary. Note that the Dictionary class is provided as a standard library class within the OOP environment. Finally, the InferenceObjects subclasses, Assertion and Rule, are instantiated to describe the meta-knowledge that is stored in the slots of the GKB frames. These frames are stored in an instance variable of the GKB object [2].

A brief example of the DKBMS objects are depicted in Figure 7. The frames are used in the GKB object to organize and focus the planning meta-knowledge for the GKB inferSubGoal method. In the example shown, the knowledge associated with "Proposals" is organized within one frame. The meta-knowledge stored within the slots of this frame will derive one or more subgoals during the inference processing. These subgoals are collected, as part of the knowledge engineering process, into knowledge clusters. The granularity of these clusters relative to the frame subgoals is subject to the judgment of the knowledge engineer. The relationship between the clusters and the subgoals is stored within the Cluster objects, and
the Cluster objects are assigned to specific KBSs in the ClusterDict object. The LanguageDict object is used in the mapping module of the DKBMS architecture to translate the communication protocol arguments from one language to another.

Each local KBS will represent its knowledge using an appropriate paradigm; however, the local KBMSs must provide some mechanism to send and receive messages conforming to the standard protocol. We propose that the C++ language be used to implement a uniform network interface control program. The C++ language was selected for its support of object-oriented programming and for its portability across incompatible computing platforms [27, 28, 30]. The local KBMSs would customize the network interface program at their individual sites to issue system calls to their respective knowledge bases and databases. Thus, any knowledge representation paradigm, commercial or customized, can be accommodated at the local sites. Standard interfaces would be supplied for common commercial knowledge-based development tools.

4.4 Integrating the DKBMS and KBS

Refer again to the research planning goals that are listed in Tables 1 and 2. In a distributed system, the goals in Table 1 would describe possible subgoals in the DKBMS that would be delegated by meta-rules to one or more KBSs. Each of these subgoals would be assigned to a knowledge cluster and that knowledge cluster would be assigned to the KBS(s) which are able to resolve goals in a similar domain. The goals listed in Table 2 describe goals that would be directed at the DKBMS, which would send the necessary subgoals to one or more local KBSs.

A dialogue using the standard communication protocol may occur between the DKBMS and one or more KBSs in order to resolve a goal. For example, if Goal 1 of Table 2 were presented to the DKBMS, then the following subgoals, via appropriate protocol messages, might be sent:

1. Human operator inputs goal of total research support dollars.
2. DKBMS identifies potential participants from industry and university departments.
3. DKBMS sends a message to KBSs in XYZ Corp and Government Agency requesting research interests and dollar support available.
4. DKBMS sends a message to KBSs in MIS and EE departments requesting research topics and desired grant dollar amount.
5. DKBMS matches MIS and EE topics with the interests returned from the funding sources, and totals the potential research dollar support.
6. Iterate over this dialogue process until the financial objectives stated in step 1 are satisfied, asking the KBSs for secondary choices.

5 Research Issues

Our DKB architecture requires the integration of knowledge bases and databases, however, discussion in the previous sections has primarily focused on the knowledge-based component. Knowledge distribution presents the most difficult unresolved problem requiring solution prior to implementing a successful prototype of this architecture. We have referred to a Knowledge Base Management System, or KBMS, which controls the integration of data and knowledge at each individual knowledge-based system (KBS) site. Research into the KBMS functionality is currently being addressed by others [9, 16, 17, 21, 33]. Our architecture allows the database at a KBS site to be
accessed directly through an existing DBMS, however, when data is required by a distributed knowledge-based application, the database will be accessed through the KBMS at each local site. Thus, if the DKBMS requires data-based information while resolving a global goal, it would utilize the communications protocol to send a message to the local KBMS responsible for the required data. If the data were also distributed, then the KBMS would communicate with a DDBMS to retrieve the data.

The proposed DKBMS architecture leaves several issues for further research. Although we have presented an object-oriented definition for the global knowledge base, this choice does not eliminate alternative forms of representation. The most appropriate representation will likely depend upon the application domain requirements; our future research will attempt to provide design heuristics for making this decision. The inference engine employed by the GKB deserves significant research attention to design and apply the meta-knowledge.

The appropriate determination of knowledge clusters, and allocation of local KBSs to these clusters, will also have a significant impact on the performance and flexibility of an application using this architecture.

An additional research issue is to determine the location of the DKBMS itself. The DKBMS may be located entirely at one site of the distributed network, duplicated at every network site, or selectively decomposed by distributing the DKBMS components. For example, if the frames that were described in the last section to represent the GKB meta-knowledge could be split regionally or functionally between several KBS sites.

Several data communications issues require investigation for the DKBMS architecture. A policy must be defined for acknowledging the communication protocol messages sent between the KBMSs and all KBSs. A related issue is the system response time required to resolve global goals; communication between sites must be minimized for real-time applications of the DKBMS architecture. The DKBMS must maximize reliability of the overall system by ensuring that the global knowledge base is always available to the network.

6 Conclusions

We believe that this Distributed Knowledge-Based System architecture can improve the effectiveness of information management by facilitating the design of information systems which support a distributed group of semi-autonomous decision makers. This group decision process requires a network of databases and knowledge bases with coordinated communication among them. Cooperative action needs to be spurred, especially in situations where the decision makers have conflicting subgoals. Performance of the integrated information system may be enhanced by allowing parallel processing of sub-tasks by individual knowledge-based systems. When cooperative decisions are required, each knowledge-based system must have access to multiple external sources of data and knowledge.

Our research synthesizes the work from several related areas, including distributed database systems, distributed problem solving, and object-oriented design. Previous work in Knowledge Base Management Systems has focused on integrating knowledge and data at a single site. This integration is necessary for the architecture that we have described, however, many business applications require data and knowledge to be integrated from logically and geographically distributed sites. Each local site has a set of objectives implicitly incorporated into its individual knowledge-based system, but the independent sites must cooperate to produce a coherent plan which maximizes the global objectives.

The DKBMS architecture specifically supports sharing and integrating knowledge and data within the global system and allows heterogeneous knowledge-based systems to communicate through the DKBMS via a standard protocol. Generic queries, from users with system-wide goals or users having no knowledge of the network organization, will be processed by the DKBMS. The DKBMS itself may be centralized at a single node, or distributed on a few or all nodes. Thus, the DKBMS architecture can be implemented with various degrees of centralization or decentralization. Partial centralization of control is desirable in many business applications to facilitate maintenance within a dynamic network structure.

A prototype DKB for research planning is being developed to demonstrate the feasibility of the DKB architecture functionality and to resolve or refine the research issues described above. Through this prototype, we will define the communication protocol required to minimize network communications overhead. DKBMS distribution policies will also be explored to evaluate multiple degrees of centralization. Our eventual research goal is to provide recommendations for using the DKBs architecture in a wide variety of business application domains.

7 References


63