

객체지향 지식베이스(OOKS)에서의 절차적정보 표현과 활용

Representation and Facilitation of Procedural Data in OOKS: a Knowledge Base Model Using an Object-Oriented Database

허순영* · 양근우* · 최지윤**

* 한국과학기술원 테크노경영대학원

** 대림정보통신

Abstract

Building a knowledge base effectively has been an important research area in the expert systems field. A variety of approaches have been studied including rules, semantic networks, and frames to represent the knowledge base for expert systems. As the size and complexity of the knowledge base get larger and more complicated, the integration of knowledge bases with database technology becomes more important to process the large amount of data. However, relational database management systems show many limitations in handling the complicated human knowledge due to its simple two dimensional table structure. To overcome these limitations of relational database systems, we proposed Object-Oriented Knowledge Store (OOKS), a knowledge base model on the basis of a frame structure using an object-oriented database. In the proposed model, managing rules for inferencing and facts about objects can be successfully performed through one uniform structure, knowledge and data can be tightly coupled and the performance of reasoning can be improved. For building a knowledge base, a knowledge script file representing rules and facts is used and the script file is transformed into a frame structure in database systems. Specifically, through designing a frame structure in the database model as it is, OOKS can facilitate management and utilization of knowledge in expert systems.

The knowledge base architecture of OOKS has the following subcomponents to enable users to represent and facilitate rules and facts in expert systems knowledge bases. The script parser interprets the contents of the script file created by a knowledge engineer and the knowledge load manager is in charge of saving the knowledge such as rules or facts either interpreted by the script parser or resulted from inferencing into the database. The script generator performs the opposite function of the script parser. It assembles the frame-based knowledge contents in the

database and recreates the script file for users to view the actual knowledge in the database in a familiar form. Additionally, the user interface connects the expert system users and the inference engine, which makes the inferencing strategies and performs the actual inferencing using the knowledge such as rules and facts stored in the database and input from the users. Through the user interface, the user can enter the additional information needed for inferencing and the inferencing results can be displayed. The knowledge retrieval manager searches the knowledge base for the needed information for inferencing and transfers the searched knowledge to the inference engine. Finally, the virtual memory mapping manager plays an important role in improving the inferencing performance by moving the requested knowledge from the servers database to the clients virtual memory which can make it faster for the knowledge retrieval manager to access the database.

Procedural data, which can execute a course of actions when predefined conditions are met, can be used to perform numeric or statistical calculations, manipulate knowledge bases, and set integrity constraints if properly represented and facilitated in knowledge bases. Using external function calls to implement procedural data features in expert system shells has disadvantages such as requiring programming knowledge from expert system users and having possibilities to cause integrity-related problems to relational databases.

To overcome those limitations of external function calls and provide better procedural data management, this paper extends the knowledge base architecture of OOKS. The improved architecture of OOKS can represent not only facts and rules but also procedural data for automating needed processes according to the changes of the knowledge base through the operation definition feature of object-oriented databases. To ease the representation and utilization of procedural data in knowledge bases, user script command structure is devised based on Event-Condition-Action rule knowledge model of active databases. Also required object class structure in an object-oriented database to facilitate procedural data is proposed. The user script commands were added to represent and facilitate the procedural data for OOKS and OOKS follows the same steps to process the procedural data as it does with rules and facts. A knowledge engineer defines the procedural data associated with an appropriate fact using script commands and each subcomponent of OOKS performs its role to represent and execute the defined procedural data when predefined conditions are met.

To test the appropriateness of the proposed knowledge base model, a prototype system has been developed using a commercial ODBMS called ObjectStore and C++ programming language on Windows NT environment. Through the prototype system, it is verified that not only the expert knowledge in the form of rules, facts, and procedural data can be effectively stored in an object-oriented database but also the inferencing using the proposed knowledge base structure outperforms that of traditional systems using relational databases. Using the implemented prototype system, rule-based inferencing such as backward or forward inferencing can be performed based on the knowledge represented as a script file and stored in the knowledge base. Throughout the paper, the famous World Traveler example by Hull was used to show the structure and usage of OOKS.