HEXPERT: An Expert System Building Tool

Suk I. Yoo, I. K. Kim, C. H. Park, H. J. Chang, T. G. Kim, and M. K. Min

Abstract

This paper presents a tool for building expert systems, called HEXPERT, which has been designed and implemented by our lab. The HEXPERT has been developed by integrating several techniques and ideas known to relax the difficulty of development of expert systems and to enhance the performance of expert systems. The HEXPERT provides the environment where the knowledge base is formulated from several sub-knowledge bases each of which is represented using an object-oriented programming technology and processed in any of three inferring directions, the forward, the backward, and the mixed direction. It also provides two browsers, called the object browser and the inference browser, which enable the knowledge engineers to develop and maintain its application systems easily.

To transfer data between expert systems and database management systems, the HEXPERT can communicate to the INGRES database management system. Finally, as one of application systems developed using the HEXPERT, an expert system for diagnosing automobile engines is introduced.

1 HEXPERT

The HEXPERT (Hyundai-EXPERT) is an expert system shell which has been developed using C language on Sun workstation. As shown in Fig. 1, the HEXPERT consists of five modules, a knowledge base, an inference engine, a user interface, a database interface, and an object module.

Knowledge Base: One of the difficult problems in developing expert systems is to formulate the knowledge base from the knowledge source and update it when necessary. This problem becomes more serious when the amount of knowledge about the applied domain is very large. To resolve this problem, a method to partition the overall knowledge into several sub-knowledge according to independent or semi-independent tasks performed in the domain, represent each sub-knowledge by one sub-knowledge base, and integrate all sub-knowledge bases by resolving the conflict if exists between sub-knowledge bases may be suggested. Further if each sub-knowledge base is represented using an object-oriented programming technology and allowed to be processed in any of three inferring directions, the forward, the backward, and the mixed one, the knowledge about the domain may be efficiently represented.

A knowledge base of the HEXPERT consists of rulebase and factbase. Rulebase is designed to support the concept of sub-knowledge bases by introducing RULECLASSes. Each RULECLASS containing rules formulates one sub-knowledge base and declares one of the three inferring directions to process it. The following example is one of RULECLASSes defined for the DIAS1 introduced in the next section.

```
RULECLASS AFS.m[ENGINE]
{
    DIRECTION: forward
}

RULE afs.1 (AFS.m)
{
    if: (car)
    do: (print "Does engine start?(On/Off)\" ")
    (read en-st)
    (modify car start = en-st)
}
```

Factbase declaring the types and the default values of the attributes of objects or the active image of objects used in the rules is designed to support an object-oriented programming technology by introducing FACTCLASSes, FACTs, and METHODS. Each FACTCLASS stands for one object class with the types and the default values of the associated attributes of the object class and each FACT represents one instance of one object class by specifying the re-
lated values of the attributes defined in the object class. Each METHOD gives an algorithm realizing the active image of the associated objects.

Property inheritance, which is well known to be one of the advantages of an object-oriented programming, can be done between two FACTCLASSes by specifying one FACTCLASS as the superclass of the other FACTCLASS or between a FACTCLASS and a FACT by specifying the FACTCLASS as the class of the FACT.

Inference Engine: An inference engine is designed to support three different inferring directions, the forward, the backward, and the mixed one. Any of the three directions can be declared in each RULECLASS as explained before.

It is well known that the efficiency of the inferring process is greatly dependent on the matching algorithm used for finding the necessary fact or rule from the knowledge base. In the HEXPERT, more than two matching algorithms for the forward and the mixed inferring directions have been developed based on the Rete network algorithm[1]. Since the Rete however was developed for processing the simple rule-based knowledge in the forward direction, it can not be directly applied to the object-oriented knowledge. To process the object-oriented knowledge in the forward direction, the Rete has been extended by adding the so-called structure-node and filter-node to the node network generated by the Rete. To process it in the mixed direction, the Rete has been extended by adding the so-called hypothesis-nodes to its node network. Finally, the matching algorithm used for the backward direction has been developed based on the backtracking algorithm using stack structure.

User Interface: A user interface provides two browsers, called the object browser and the inference browser. The object browser consisting of the factbase browser and the rulebase browser is designed to allow the knowledge engineers to edit the knowledge base easily using this. The factbase browser shows the hierarchical structure of FACTCLASSes and FACTs defined in factbase. By activating the necessary FACTCLASS or FACT, the knowledge engineers can access its contents and do any operation like insertion, deletion, and update. Similar operations are allowed in the rulebase browser. The inference browser is designed to allow the knowledge engineers to check if the systems being developed are correctly working during the inferring process. It shows a sequence of rules fired during the inferring process which results in the current conclusion. By activating one of several functions provided by the browser, the knowledge engineers can access the associated part of the sequence and do the necessary actions for achieving the desired result.

Database Interface: A database interface is designed to provide two operations, UPLOAD and DOWNLOAD, for data transfer between factbase in the HEXPERT and database in the INGRES database management system. The UPLOAD operation which transfers data from factbase to database is implemented by converting one table and its tuples into one FACTCLASS and its associated FACTs. Conversely, the DOWNLOAD operation which transfers data from database to factbase is implemented by converting one table and its tuples into one FACTCLASS and its associated FACTs.

Object Module: An object module provides a set of operations defined on factbase of the knowledge base. Since the operations provided by the object module are shared by all the transactions related to factbase during the inferring process, the overall performance of the system is enhanced. The detailed description of the HEXPERT can be found in [2].

2 Application

As one of application systems developed using the HEXPERT, we briefly introduce an expert system for diagnosing automobile engines with ECUs, called DIAS1. The DIAS1 diagnoses two types of automobiles produced by the Hyundai Motors Co. in Korea: automobiles with self-diagnosis mechanism and automobiles without self-diagnosis mechanism.

The knowledge taken for diagnosis is formulated by four interrelated rulebases to be processed in the forward inferring direction. One of the four rulebases, called ENGINE, contains 12 ruleclasses with 240 rules which represent the knowledge for diagnosing the components related to the ECU. Rulebases, called TESTER and DATA, contain one ruleclass with 30 rules and one ruleclass with 80 rules respectively which process the input data for diagnosing each of the two types and activate the associated rule contained in ENGINE. Finally, rulebase, called MPI, contains one ruleclass with 4 rules which ask the types of automobiles diagnosed and activate one of the two rulebases, TESTER and DATA, according to the given automobile type. For each RULECLASS defined in rulebase, one or more than one FACTCLASS is defined to declare the values or the types of the attributes of the object used in the rules contained in it. For each FACTCLASS, one FACT is defined to accept the current status of the object given during the diagnosing process. Factbase designed for rulebase of the DIAS1 is formulated on the 4-level tree structure. The details of the DIAS1 can be found in [3].

Acknowledgements

The authors thank Mr. M. H. Chung, the president of the Hyundai Electronics Industrials Co., for his financial support and encouragement on the project for developing the HEXPERT and the DIAS1.

References

