Domain Model for Constructing a Knowledge Based System


*Energy Research Laboratory, Hitachi, Ltd., 1168 Moriyama-cho, Hitachi-shi Ibaraki-ken, Japan 316 Tel. (0294) 53-3111
**Software Development Center, Hitachi, Ltd., 549-6, Shinano-cho, Totsuka-ku Yokohama-shi Japan 244 Tel. (045) 821-4111

Abstract

We have proposed and developed an expert system tool ASPROGEN (Automatic Search Program Generator). It has a domain model editor, by which tool users can define their problems in an object-oriented way, and a built-in automatic generation function of a domain specific inference program. This function is based on search-based program specifications and an abstract data type of search. We show the descriptiveness of this domain model by applying ASPROGEN to a scheduling problem.

1. ASPROGEN

We have developed an expert system tool ASPROGEN (Automatic Search Program Generator). Figure 1 shows an overview of ASPROGEN. ASPROGEN has no embedded inference mechanism. Instead, it has a search program generating mechanism which produces inference programs according to user specifications of the search program, domain knowledge, and detailed constraints.

We have already introduced and presented the ASPROGEN's generating mechanism of inference program[4]. ASPROGEN users develop expert systems by the following procedure.

Step 1: Users specify a problem solving strategy from the viewpoint of a search strategy.
Step 2: Users input the search strategy by selecting the classification items of the search classification tree which the tool prepares. This step is executed with the help of the tool interface.
Step 3: Users input domain knowledge and constraints with the help of the tool interface.
Step 4: ASPROGEN generates a domain specific inference program and converts data structures for domain knowledge into C source code.

Step 3 corresponds to the modeling process of domain knowledge and we focus our discussion on step 3 in this paper.

2. Description of a problem by a domain model

2.1 Generated inference program

ASPROGEN generates a global inference program according to the specified problem-solving strategy and local inference programs from the described domain knowledge and constraints.

The generated global inference program is the tree search program, and the inference process can be regarded as an allotting value process. To decide attribute values of the search node, constraints and domain knowledge concerned with the node attributes must be referred to.

Thus, the domain model of ASPROGEN must bridge the gap between global and local inference programs.

2.2 Describing domain knowledge

ASPROGEN has an interface for describing domain knowledge which is described by objects and attributes, attribute value ranges, and attribute constraints. There are two types of objects. One is a class object which defines attributes and relations between other objects. The other type is an instance object which has instantiated attribute values.

Figure 2 shows a representation scheme for domain knowledge in ASPROGEN. Nodes of the search tree are also objects. Node objects are related to other objects. By connecting this search node object and domain object,
domain knowledge and the problem solving strategy are related to each other. Therefore domain knowledge can be used in the process of problem solving. Objects are related in three ways.

1. Class-instance relations: Instance objects have the same attributes as class objects, and the values of the attributes are inherited from the class objects.
2. Attribute-value relations: The value region of the attributes can be described by the class objects. Thus, the attribute value region is a set of instance objects of class objects.
3. Attribute-object relations: The attributes of the objects can be described by class objects. Thus, the attributes of the nodes are instance objects of the class objects, and attribute values are those of the instance objects.

\[ \text{Attribute-object relation} \rightarrow \text{Attribute-value relation} \rightarrow \text{Class-instance relation} \]

Fig. 2 Scheme of knowledge representation in ASPROGEN

On the basis of these definitions of domain knowledge, ASPROGEN users describe them. ASPROGEN provides a simplified language which can describe constraints by using object names and attributes.

2.3 Example

We examine the descriptive power of domain model of ASPROGEN by applying it to a scheduling problem. Inputs of the jobshop scheduling system are identification number of order, product name, production amount, the delivery date, and so on. The outputs include machine identification number to be used for each order, and starting and ending time of machine operation for each order. The objective function is the machine operation rate. Figure 3 shows a domain model for the example. Attributes of the search node are starting and ending times of machine operation for each order, and by setting appropriate values, search node as a schedule plan is generated. Other objects include those of orders, machines. Objects of orders have attributes, such as identification number, product name, production amount, machine to be used, and so on. The constraints of machine objects are available production and machine setting time.

ASPROGEN generates constraint satisfaction programs for each constraint from the kernel of the constraint satisfaction programs, and generates candidate values for each attribute value.

\[ \text{Search tree node} \rightarrow \text{Orders} \rightarrow \text{Orders_1} \]

\[ \text{Name} \rightarrow \text{Identification No.} \]

\[ \text{Order[n] starting time; int n < 1000} \]

\[ \text{Importance} \rightarrow \text{The date of delivery} \]

\[ \text{Product name} \rightarrow \text{Production amount} \]

\[ \text{Machine to be used} \rightarrow \text{Machine_1} \]

\[ \text{Identification No.} \rightarrow \text{Type of machine} \]

\[ \text{Occupied time intervals} \rightarrow \text{Available production} \]

\[ \text{Machine setting time} \rightarrow \text{attributes} \]

\[ \text{constraints} \rightarrow \text{objects} \]

Fig. 3 Domain model for job-shop scheduling problem

This example includes various kinds of constraints and it is easy to apply the domain model to another kind of problem by defining objects, attributes, and constraints properly.

3. Conclusions

We have proposed and developed an expert system tool ASPROGEN (Automatic Search Program Generator) in which the automatic generation function of a domain specific inference program was built in. ASPROGEN has interfaces for domain knowledge using an object-oriented approach and constraints which represent control knowledge. It is described by using domain knowledge and can cover a detailed problem solving strategy.

We applied ASPROGEN to produce a scheduling system. This work showed domain model of ASPROGEN has enough coverage for complicated, structured problems.

References