TR-227

FACT/MODEL REPRESENTATION ENVIRONMENT IN AN EXPERT SYSTEM TOOL ON PSI

by

H. Kubono, J. Sawamoto,
Y. Nagai and Y. Iwashita

January, 1986

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

FACT/MODEL REPRESENTATION ENVIRONMENT IN AN EXPERT SYSTEM TOOL ON PSI

Hideo Kubono, Jun Sawamoto, Yasuo Nagai and Yasuo Iwashita

Fifth Research Laboratory, ICOT, Tokyo, Japan

ABSTRUCT

This paper describes the architecture of our prototype for development of a sophisticated tool capable of facilitating construction of expert systems in a wide variety of domains. This prototype is called PROTON. The architecture of PROTON is based on the logic and object oriented programming language, Extended Self-contained Prolog (ESP) impl$\overset{e}{i}$mented on the personal sequential inference machine (PSI).

The architecture of the environment for representing facts or models in this prototype is the focus of this study. This environment is called the fact / model representation environment (FME).

1. INTRODUCTION

Expert systems are one of the most promising application fields of AI technology. Their effectiveness has been demonstrated in several fields, and recently a lot of hybrid expert system building tools have appeared. But the practical advantages of their architectures are not yet clear. While studying the architecture at the abstract level, we are pursuing research of the architecture of a general purpose tool for expert systems through case study. We are developing as the environment for both tool architecture research and for the case study itself. This tool is also an experimental system for demonstrating the results of the FGCS project.

2. OVERVIEW OF PROTON

PROTON is a hybrid knowledge representation environment realized by utilizing ESP's object oriented and logic programming facilities. This environment has the three following categories for representing knowledge.

(1) Fact / model environment -FME-

This sub-environment is called FME. In FME knowledge about components of the problem domain is represented in the form of frames. Knowledge in FME is referenced or modified by the commands issued from knowledge sources. The mechanism will be explained in detail later.

(2) Heurstics environment - Rule Base -

This consists of multiple units, each of which is called a knowledge source (KS). Each unit includes rules for forward or backward reasoning and information which defines the manner of firing rules. Each unit corresponds to a subtask for solving a given problem. Thus problem solving knowledge can be efficiently modularized, and from the implementation point of view, this approach makes it possible to execute rules efficiently because search of rules in any given case is considerably restricted. To provide smooth incorporation of an expert system user interface typical for consultation or diagnosis systems, explanation (such as Why and How) and ask-user facilities are embedded in the rule base. The user-defined functions in ESP could be included in the system, and fragments of ESP code including these function calls could be mixed in rules for good expressive flexibility.

(3) KS control environment - Metaknowledge -

Metaknowledge is knowledge about control of the KS mechanism. It specifies the activation timing of the KS mechanism in the form of forward chaining rules called META RULEs. Metaknowledge is represented in a unit separate from the KSs. By monitoring the information contained in the working memory of FME, this kowledge can determine the status or phase at which a problem solving task stands, and decide the appropriate KS to be run next. Describing metaknowledge in rule format makes it easy to realize various mechanisms to execute KS activations. For example, it can desribe nondeterministic control of KSs. If a user extends the flexible metaknowledge function, a blackboard model architecure, like that of Hearsay-II, could be easily designed. Cf. Fig. 1

This prototype tool has the following additional feature.

Statistical Data Extraction

This facility allows extraction of some statistical data, e.g., the locality of references for the working memory. The data is used for optimizing organization of the KSs.

These knowledges described in an external format which PROTON provides are compiled into ESP codes for execution.

3. FACT/MODEL REPRESENTATION ENVIRONMENT(FME)

FME is an environment which consists of an external representation language for describing facts or models and a group of system functions manipulating them.

Facts or models are located in a working memory provided by FME. These facts are instance elements instantiated from the templates which should be defined in advance. Working memory management system (WMMS) accepts access requests from rules. WMMS carries out manipulations of instance elements like creation, deletion, modification, and various searches, such as a search for an instance element having a certain attribute's value. The relation mechanism incorporated here is completely user-defined and manipulable by rules. Cf. Fig.2

In many expert systems, frame style representation is employed because of its advantage of making structural facts or models in a problem domain transparent. FME also employs this, not only for elemments of fact/models, but also for relations between them.

Facts in a problem domain accessed by rules are represented in frame style in this environment. Frame representation with IS-A and HAS-PART hierarchies and attached procedures are realized by utilizing the mechanism of ESP. To define the structure of elements, the templates, called template of element (TE), should be described in the external representing language provided by FME. A TE mainly consists of several attributes defining its characteristics, and each attribute consists of a group of facets. When an elemment-instance is instantiated from the TE, these inherited attributes acquire particular values.

Definitions of relations between elements can be represented as frame-type elements enabling users to describe relations among multiple elements easily. Also, to represent relation-instances, the template, called template of relation (TR), should be prepared in advance for each type of relation. A TR mainly consists of the information about restrictions on the arguments corresponding to TE elements, logical combinations of other TRs, called SUBSTITUTIONs, and equations called INTERPRETATIONs. When a relation-instance is instantiated from a TR, it takes the form of a proposition with names of related elements as arguments from the outer FME. Cf. Fig.3

Some specific features and capabilities of FME based on the above are described below.

3.1 INHERITANCE HIERARCHY

The inheritance mechanism operates when representing the hierarchy of the conceptual abstraction defined at TEs and TRs. With respect to TEs, information about each attribute is inherited from the superior abstract TE for its facets. With respect to TRs, information mainly about SUBSTITUTION and INTERPRETATION is inherited from the superior abstract TR. This multiple inheritance can never be manipulated after loading TEs and TRs in WM.

3.2 MECHANISM FOR WHOLE-PART HIERARCHY

This is the hierarchy of the abstractions specifying some model. The whole-part relation is realized by linking between the whole-element and the part-elements, so this relation is updated dynamically during inference. Every element linked is made from some TE. Information about constituents is managed by their direct superior element. When retrieval of a model's part is required, this model searches it automatically. Cf. Fig.4

3.3 MECHANISM FOR MAINTENANCE COHERENCE

Relation-instances loaded in WM are used not only for describing the relations between elements of facts, but also for maintaining the relations semantically by using its INTERPRETATION embedded in the relation-instances. INTERPRETATION is a kind of demon which mantains semantic coherence about facts or models in WM, by producing some side effect at the attribute level, over the elements linked by the relation-instance.

In INTERPRETATION, each term is the attribute's name of an element-instance corresponding to one of the arguments. When one of the terms is modified at the element-instance including it, this element-instance sends a request asking every related relation-instance to activate its INTERPRETATION, then INTERPRETATION manipulates the values of other terms at these corresponding element-instances by using FME commands. This process can activate other INTERPRETATIONs. So the side effect generated by updating one element-instance in WM can propagate to all of the related instances through relation-instances. Thus the semantic coherence of facts or models in WM can be maintained dynamically during inference by the KSs.

In this proccessing, if the chaining of this side effect is permitted only by instances, it is feared that the reaction can never be terminated and the maintenance cannot succeed. The chain reaction is monitored by WMMS to avoid this loop. Cf. Fig.5

3.4 MECHANISM FOR RELATION SEARCH

When the relation-instance to be retrieved is not found, the TR able to instantiate it can translate the logical combinations of relations semantically equal to the concerned relation, by using the SUBSTITUTION described in it. These combinations consist of the commandsfor retrieving relation-instances. Furthermore if a command in the sequence fails, it is unfolded into another combination of relations semantically equal to it. Thus it can be proved whether the relation is logically concluded.

If this search of relations is left only to the TRs, it is feared that the process cannot be terminated. So the searching process is monitored by WMMS, and WMMS cuts off the search paths which would generate the infinite search. Cf. Fig.6

4. EXTENDED FACILITY OF FME

- Synthesis and verification of communicative models -

The FME environment can be easily extended to realize a model of a communicative network by using the advantages of the primitive functions of FME previously mentioned. It is aimed not only at estimating the functions of FME, but also at investigating the characteristics of synthesis and diagnosis.

In this environment, users or KSs including forward chaining rules can sythesize the complicated or concrete communicative model from simple or abstract specifications. The synthesized model is represented as a hierarchy of the abstraction levels. Furthermore, a hierarchical communication network is realized in this model. The communicative processing in this network is realized by employing the INTERPRETATION function of TR-instances. This TR is created as a primitive unit in advance. The entity instantiated from this TR is called MEDIA, and communication links realized by MEDIA are called MEDIA-LINK.

The synthesis and verification processes of communicative models are explained below.

4.1 SYNTHESIS OF THE COMMUNICATIVE MODEL

This communicative model is a hierarchical structure of communicative constituent sub-models and HAS-PART links among them. These links are realized by a command sequence issued by rules of the KS through the FME manager. Particular attributes of the constituents must be declared in advance as ports for MEDIA-LINK on the templates. Each communicative sub-model is linked at its ports to ports of another model through MEDIA, and the communicative network is constructed in this way. Any model on any abstract level can be represented with a network of its constituents. Thus, a communicative model is represented in the form of a hierarchical structure. Cf. Fig.7

TEs are prepared for corresponding functional-block templates to provide their instances as constituents needed for model sythesis. Each instance has two main specification categories as follows.

1. STRUCTURE definition
   a. I / O restriction of each port :
      This information is set up at the TE statically. When ports come to be linked dynamically during reasoning, the validity is checked with it.
   b. Submodel information :
      Names of any model instance's direct submodels are stored and managed in it, in term of HAS-PART linkage.

Linkage information of constituents is not managed at a particular location. Signal processing control of a superior model is unnecessary because of the features of this language.

2. BEHAVIOR definition
   a. Formula calculating output values :
      Behavior, a model function mapping values at the input port to the output port, is described in the form of logic fomulas. Terms of this formula are port names of its TE. This description is written at the TE in advance.

Synthesis procedures proceed in the order below.

1. Instantiation of the top abstract model in Working Memory.
2. Matching between LHSs of rules in forward Knowledge Source for synthesis.
3. If failed, the process is terminated.
4. Generation of its submodels.
5. Linkage to submodels and among them by using HAS-PART and MEDIA-LINK.
6. Go to 2.

4.2 VERIFICATION OF COMMUNICATIVE MODEL

Data for comparison or evidence is required to verify the result of synthesis. In FME, only simulation of the communicative model makes it possible to verify its function. The simulation is as follows.

4.3 TWO-WAY SIMULATION OF COMMUNICATIVE MODEL

Simulation of the communicative model, a kind of signal processing and signal transmission, FME employs a strategy for its verification by using the hierarchy efficiently. The simulation mechanism in FME is intended to verify the model simultaneously while simulating its behavior, by two ways for signal processing.

To start the simulation, only signals can be set up at input ports of the top abstract model. When this is completed, the communicative model begins to run automatically. During simulation, signal processing of each constituent proceeds in two ways. One is the way in which the signal processing of each functional block is perfomed with its behavior descriptions. The other is the way in which the signal is sent to its subconstituents via MEDIAs and the signal processing is done by the subconstituents. If the output from its subconstituents differ from the output values calculated by its behavior definition, simulation is interrupted and the error port information is sent out from FME. Thus, verification of the communicative model can be performed with simulation.

5. SUMMARY

An outline of PROTON and the features of FME were described in this paper. This expert system tool is now being implemented and its capabilities will be evaluated by building some expert systems.

REFERENCES

[1] Buchanan, B. G., "Research on Expert Systems", HPP-81-1, Department of Computer Science, Stanford, 1981.

[2] Hays-Roth, F., Waterman, D. A., and Lenat, D. B., "Building Expert Systems", Addison-Wesley, 1983.

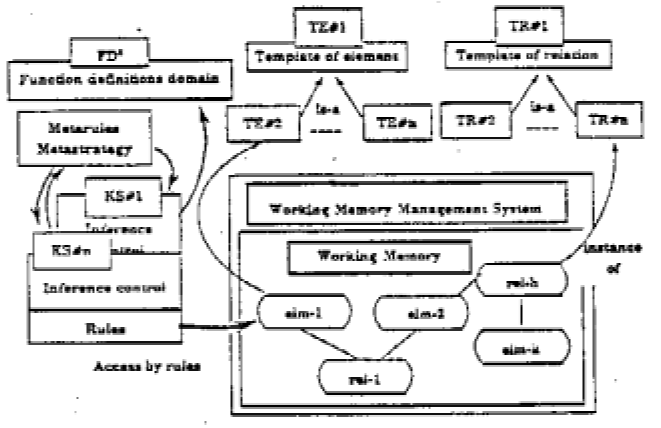

Fig.1. Structural overview of PROTON

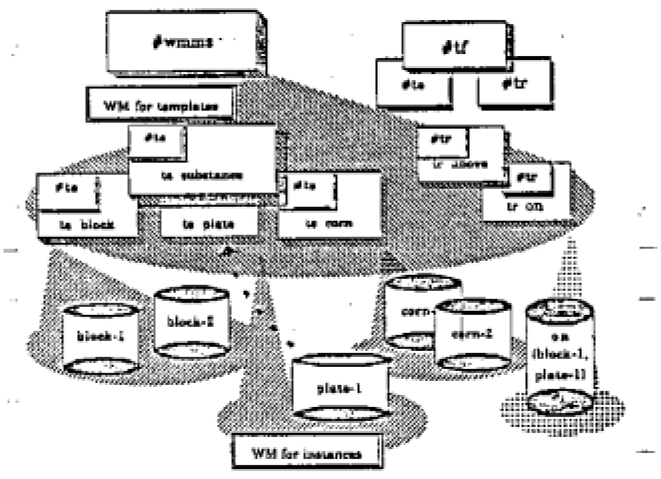

Fig.2. Working memory management system (WMMS)

```
te steel-block-with-bar                  tr on
    super                                    super
        block                                    above, contact
    attrib                                   restrict2
        material default "steel" &               (1,substance)(2,substance):
        grav default 5 &                         (1,position-z) > = (2,position-z)
        weight                               interpret    (1,position-z,new)
            if-gotten, arbit, before              <-(2,position-z,new)
            weight                                +((1,position-z,old)
            <- grav * size-x * size-y             -(2,position-z,old))
                    * size-z                 on([upper, lower])
    has-part                                     subst
        bar ( bar : (material , "steel"))    under([lower, upper])
end.                                         end.

              (a)                                  (b)
```

Fig. 3. External representation (a) Example of TE (b) Example of TR

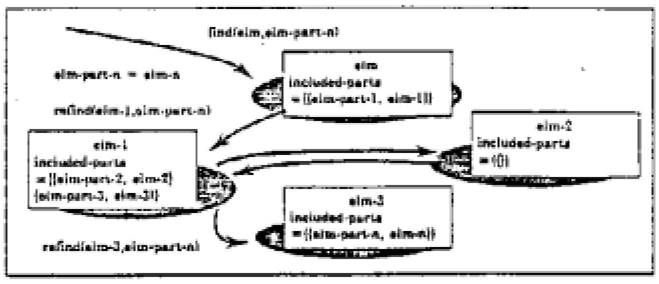

Fig. 4 Retrieval of part

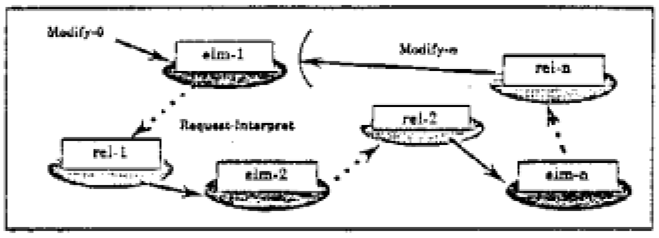

Fig. 5 Maintaining coherence

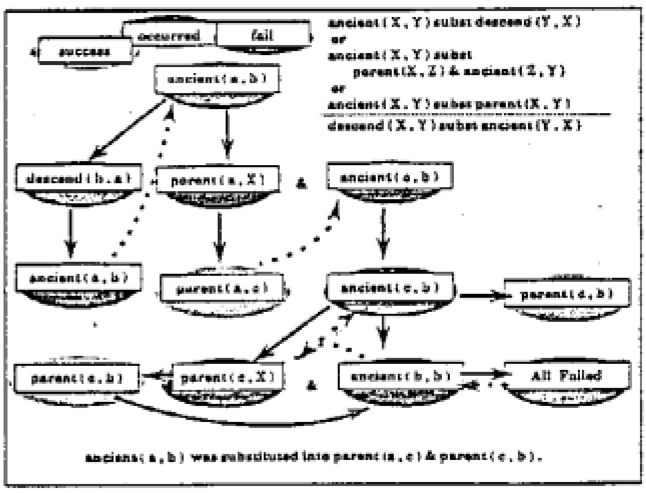

Fig. 6 Retrival of a relation using substitutive information

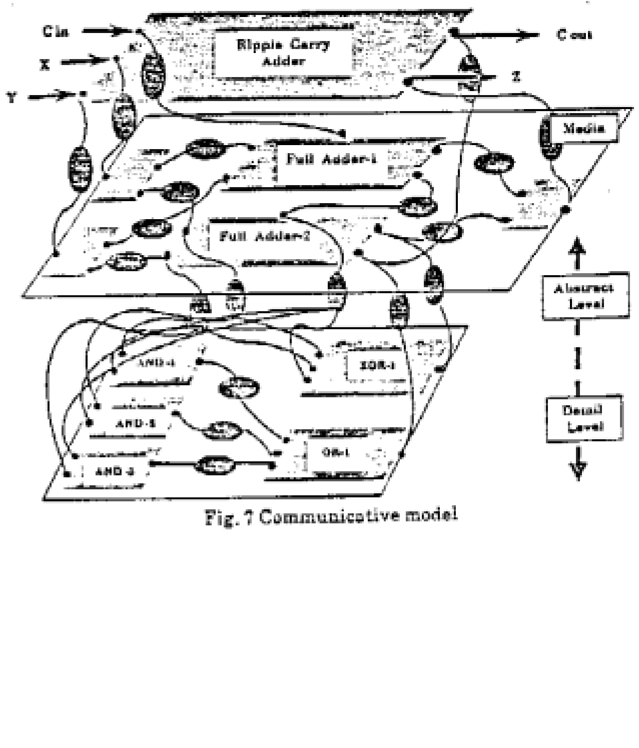

Fig. 7 Communicative model