A Model for Knowledge Based Process Control Systems

Shoichi Masui Shun'ichi Tano Motohisa Funabashi
(Systems Development Laboratory, Hitachi Ltd.)

ABSTRACT

A knowledge based control system is a process control system in which expert know-how plays an essential role in its control schemes. In this paper, a model for such a control system and a tool used to implement the system are described.

1. Introduction

A process control system based on a knowledge engineering approach is the main issue of this paper. Though progress in process automation is very rapid, the role of the expert plant operator is still important. The operator's know-how may be indispensable for smooth plant operation and the intuition of an expert operator is substantial for finding subtle abnormalties.

Substitution of such operator's knowledge with computer programs has been difficult, because conventional programming requires a fixed algorithm to represent the knowledge. Knowledge engineering[1] with symbolic reasoning, which has been applied to several practical systems[2][3][4], offers a way to make transitory operator's know-how suitable for computerization.

A knowledge based control system is a process control system which combines operator know-how with conventional control algorithms based on knowledge engineering. The way these two different kind of knowledge are combined, that is the model for combining them, is important because of the requirement for real time operation. In this paper, a hybrid model suitable for process knowledge representation is proposed. Then , EUREKA , an expert system building tool for real time operation, and its application to process control system are shown to verify the proposed model.

2. Knowledge Based Process Control System

In process control, human experts use various kinds of knowledge which include plant status derived from observed data, plant control algorithms and their experience with the plant to keep it in good condition. The knowledge can be divided into the following two types (Fig.1).

(1) process knowledge: plant state shown as observed or assumed data and the

fixed control procedure sequence for each piece of control equipment.
(2) expert knowledge: the experience, know-how, rules of thumb and problem solving strategies of each expert operator.

These two kinds of knowledge are apparently different. Process knowledge is almost fixed and specific for each piece of equipment in the process. Therefore this type of knowledge should be treated as a unit corresponding to the relevant process equipment. The representation schema of object oriented programming [5] is a suitable choice for this kind of knowledge. Each piece of equipment can be described as an object. Observed and assumed data which show states of the equipment are represented as slot-value pairs. Sequential control procedures attached to the equipment correspond to methods of the object. Control procedure s are often described in a conventional procedurallanguage such as FORTRAN, so methods should be defined in such a language.

On the other hand, expert knowledge is transitory and a characteristic of the human expert. If observed data show some abnormality, the operator may try to guess the related cause and determine appropriate control actions based on his experience or knowledge of the plant. This knowledge can be described by condition-action pairs. Therefore, the If~ Then~ rule may be a good candidate for its representation. In addition, the rule is suitable because it is convenient to add, delete or modify the toransitory part of the knowledge. In process control, it is essential to be able to explicitly manage the knowledge application sequence. Therefore, meta knowledge for controlling rule firing is necessary to express human knowledge precisely.

In summary, knowledge used in process control is categorized as follows (Fig.1):

(1) Strategic level: knowledge for controlling expert knowledge application
(2) Interpretation level: Expert knowledge for process control represented by condition-action pairs.
(3) Procedural level: fixed control procedure sequence described by a conventional language.
(4) Data level: process data represented by slot-value pairs.

3. An Expert System Building Tool: EUREKA

3.1 Knowledge representation

EUREKA is a programming environment for knowledge based process control systems. It provides IF~THEN~ rules for representing expert knowledge and objects for representing the states and functions of process equipment.

(1). Rules

Rules in EUREKA (Fig. 2) have a condition (IF) part and an action (THEN SEND) part. The condition part involves some conditions which must be satisfied for this rule to be fired. Each condition shows an object status. A fundamental condition description is represented by "(object-name $attribute-name predicate value)". For example, (pump $pressure is greater than 40) is a condition. Japanese words can be used for the description. (ポンプ の $圧力 が ４０ 以上である) is the corresponding Japanese description of the above condition. A variable, which is a name preceded by "?", can be used instead of an object-name and/or a value. When a condition is (?pump $pressure is greater than 40), any pump having pressure greater than 40 can satisfy the condition. When two or more conditions in a rule have the same variable, the variable must have the same value. The following two conditions are satisfied when pump1 and pump2 show the same pressure.

(pump1 $pressure is ?x)

(pump2 $pressure is equal to ?x)

In the action part of the rule, messages to be sent to a designated object are described. A typical message pattern is (object-name method-name (arguments list)). Thus, (?pump start (300)) is a message. This message is sent to a pump object which is specified in the IF part of the rule and invokes the "start" program with "300" as its argument. A method is a procedure attached to each object. Messages in a rule are processed in a sequential manner.

(2). Objects

Objects in EUREKA are models of process equipment. An object consists of an object-name, data part (*Data), and methods part (*Methods).

The object name should be unique in a system. In the data part, the attribute name and its value pairs are used to describe an object's status. A pump, for example, may be characterized by type, number, maximum_height, pressure, revolutions_per_minute, state, etc., each of which is an object attribute. In the methods part, names of methods(programs) attached to the object are declared. Each method itself is described using a procedural language like FORTRAN77 or C. Three methods, start, stop, and change-pressure are declared as pump methods in Fig. 3.

There are two object types, generic objects and instance objects. A generic object is a template of an instance object and holds the default value of each instance object attribute. An instance object is an active object which is referred to and invoked by rules. Each instance object specifies one generic object as its parent and inherits from it all the data and methods not

specified. A special attribute name "generic" is used for the identification of object type.

There is a super object, called SYSTEM, in EUREKA's object hierarchy. SYSTEM holds all the methods which are offered by EUREKA. As SYSTEM is assumed to be the parent of all generic objects, all instance objects can use system methods through the inheritance mechanism.

(3). Meta-rules

Rules in EUREKA can be classified into rule groups. A rule can specify its rule group name. A rule belonging to one group can befired only if the group is activated and all conditions of the rule are satisfied. Meta-rule contains knowledge about controlling rule group activation. The syntax of a meta-rule is IF (event-name) THEN (rule-group name, priority x), which means, "when the event is issued, the rule group should be activated with priority x". The rule-group is queued in an agenda memory, and the rule-group with the highest priority in the agenda memory is activated. This mechanism is useful for managing rule application and effective in representing expert knowledge.

3.2 Architecture

The inference mechanism offered by EUREKA is forward chaining. This mechanism is straightforward and suitable for the high speed processing needed for process control.

Several algorithms [6][7][8] have been proposed based on the idea of limiting the data compared with rule conditions, for example, the cross reference chart of McDermott et al.[8]. These algorithms are executed in an interpretive fashion, however, so the speed gained is insufficient for real time applications.

A new algorithm was developed for the EUREKA inference engine, based on the network matching of RETE [9]. The concepts of our algorithm are:

(1) rule conditions are translated to inner code, called a network-branch, which is a sequence of test nodes corresponding to each condition. This network-branch is connected by a newly introduced 'candidate node' to make a network tree of rule conditions. Introduction of the candidate node makes our network tree simple and easy to handle (Fig.4). This simplification saves the time and memory needed in RETE for complicated condition processing.

(2) network optimization , including merging of the test nodes and reordering a sequence of test nodes in a network-branch, is done in order to check the most difficult to satisfy test node first. This optimization cuts processing time in half.

(3) direct deletion of unused data elements stored in the network. Instead of

network processing useless elements as in RETE, an efficient pointer management mechanism for stored data retrieval is used, reducing the cost by half.

The results of the evaluation show that processing speed is independent of the total number of rules. Measured processing speed is approximately 160 rules/sec, the fastest in the world. A HIDIC V90/50 super minicomputer (2MIPS) is used for the evaluation.

EUREKA has been successfully applied to several knowledge based control and/or diagnosis systems. Among them, large scale power plant diagnosis and sem i-conductor manufacturing process diagnosis systems are in pactical use. A large scale plant, such as a power plant, has several thousand sensing points. It is not an easy task even for an expert to guess what is happening from so much data, and to determine appropriately what action should be taken in an emergency case.

In power plant diagnosis , about 200 rules are extracted from a plant expert to find the true cause of a problem. Any trouble in the system triggers rule inference and ,within 10 or 20 seconds, the true cause is reported to the operator. In this case, real time capabilities are indispensable. Therefore EUREKA is the best choice for knowledge processing. Each rule in the system is a simple one that specifies the cause by interpretation of signals comming directly from sensors.

This and other applications show that the rule and object hybrid model employed by EUREKA is easy to understand and accepted by the field expert himself, even without the aid of a knowledge engineer.

4. Conclusion

EUREKA is a rule-based and object-oriented hybrid knowledge representation tool for process control systems. Knowledge on process control is described by OBJECT, RULE and META-RULE. The OBJECT is used to represent process knowledge, that is, states and functions of the process and control equipment. The RULE represents heuristic knowledge about process control which is derived from the experience of skilled experts on the process. The META-RULE provides a way to control the sequence of RULE processing. This knowledge is used by EUREKA's forward-chaining inference engine to mimic expert behavior in process control.

Acknowledgements

The authors wish to thank Dr. Kouichi Haruna, deputy general manager of the Systems Development Laboratory, HITACHI, Ltd. for providing us the opportunity for this research.

References

[1] Barr,A. and Feigenbaum,E.A.(Eds.) : The Handbook of Artificial Intelligence (Vol 1 2 3) ; Kaufmann, 1981 1982

[2] McDermott,J. : R1:A Rule Based Configurer of Computer Systems ; Artificial Intelligence, Vol.19, 1982

[3] McDermott,J. : Building Expert Systems ; Symposium on Artificial Intelligence Application for Business, 1983

[4] Smith,R.G., : On the Development of Commercial Expert Systems ; The AI Magazine, Vol.5, No.3, 1984

[5] Goldberg,A. and Robinson,D. : SmallTalk-80 The Language and its Implementation ; Addison-Wesley, 1983

[6] Konolige,K. : An Inference Net Compiler for the Prospector Rule-Based Consultation System ; Proceedings of 6th IJCAI, 1979

[7] Mark,W. : Rule-Based Inference in Large Knowledge Bases ; Proceedings on AAAI Conf., 1980

[8] McDermott,J., Newell,A. and Moore,J. : The Efficiency of certain production system implementations : in Pattern Directed Inference Systems ;Academic Press 1978

[9] Forgy,C.L. : RETE:A Fast Algorithm for Many Pattern/Many Object Pattern Match Problem ; Artificial Intelligence, Vol.19, 1982

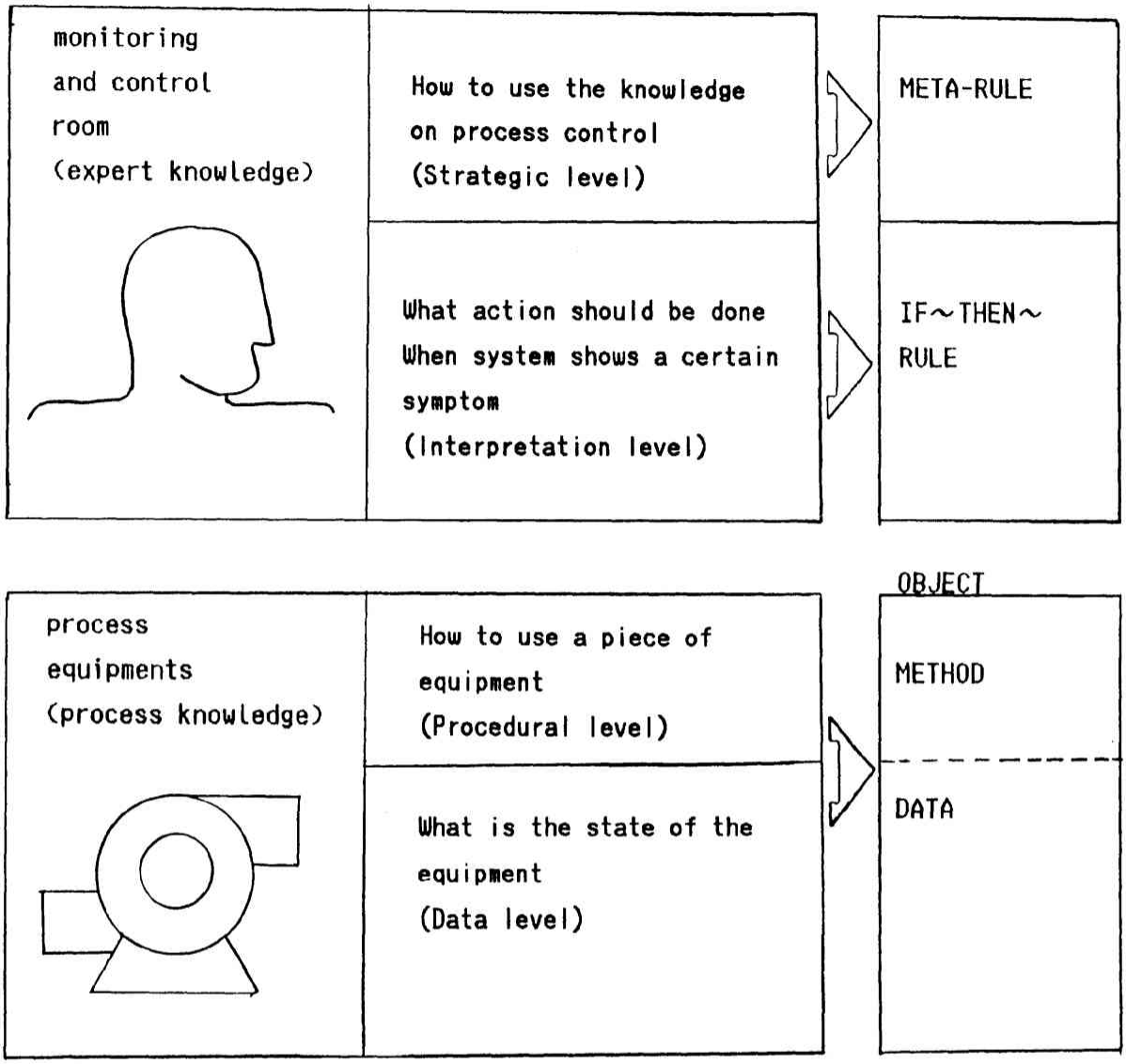
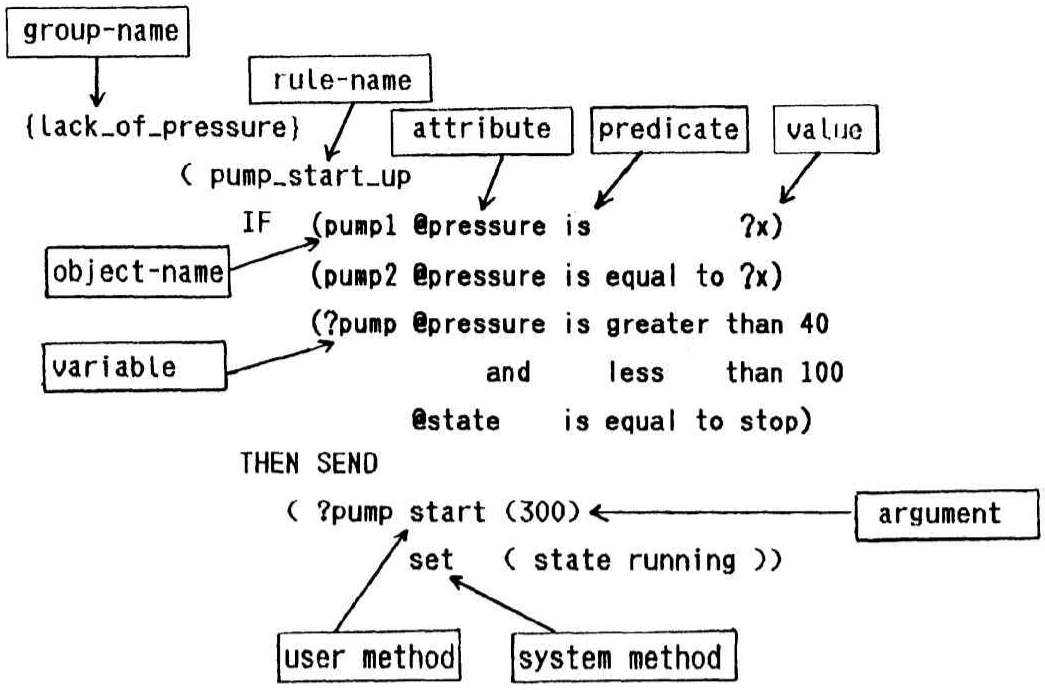


Figure 1 Knowledge Representation Schema



Meaning:

If both pump1 and pump2 have the same pressure and there is a pump having pressure greater than 40 and less than 100, then start the pump at 300 rpm.

Figure 2 Rule Representation in EUREKA

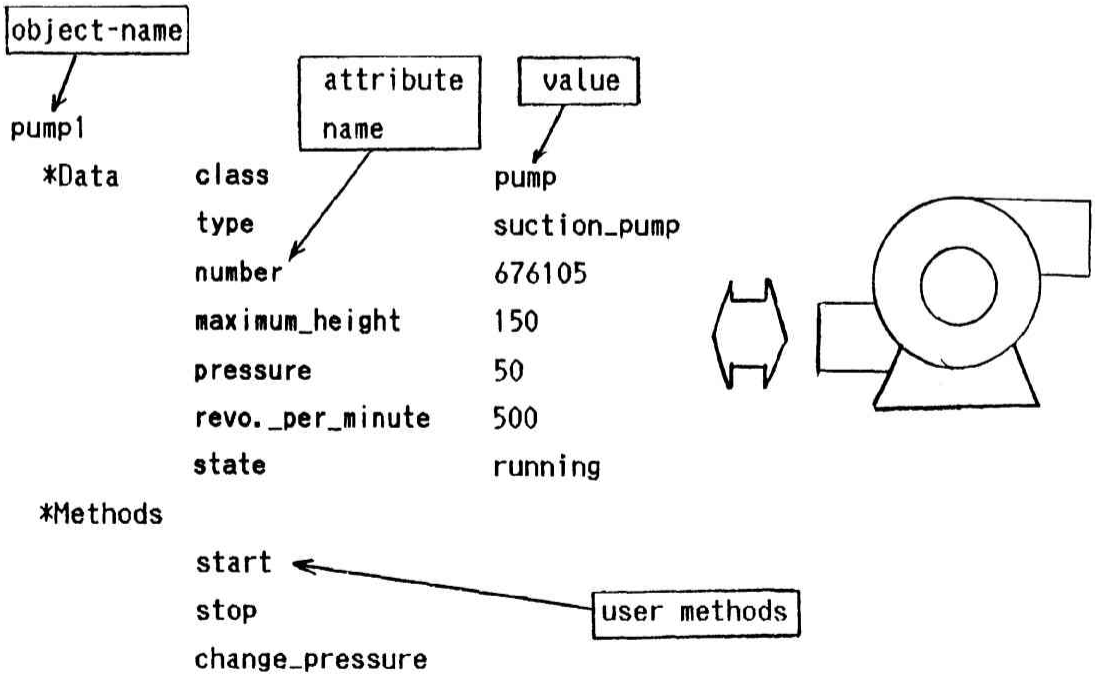


Figure 3 Object Representation in EUREKA

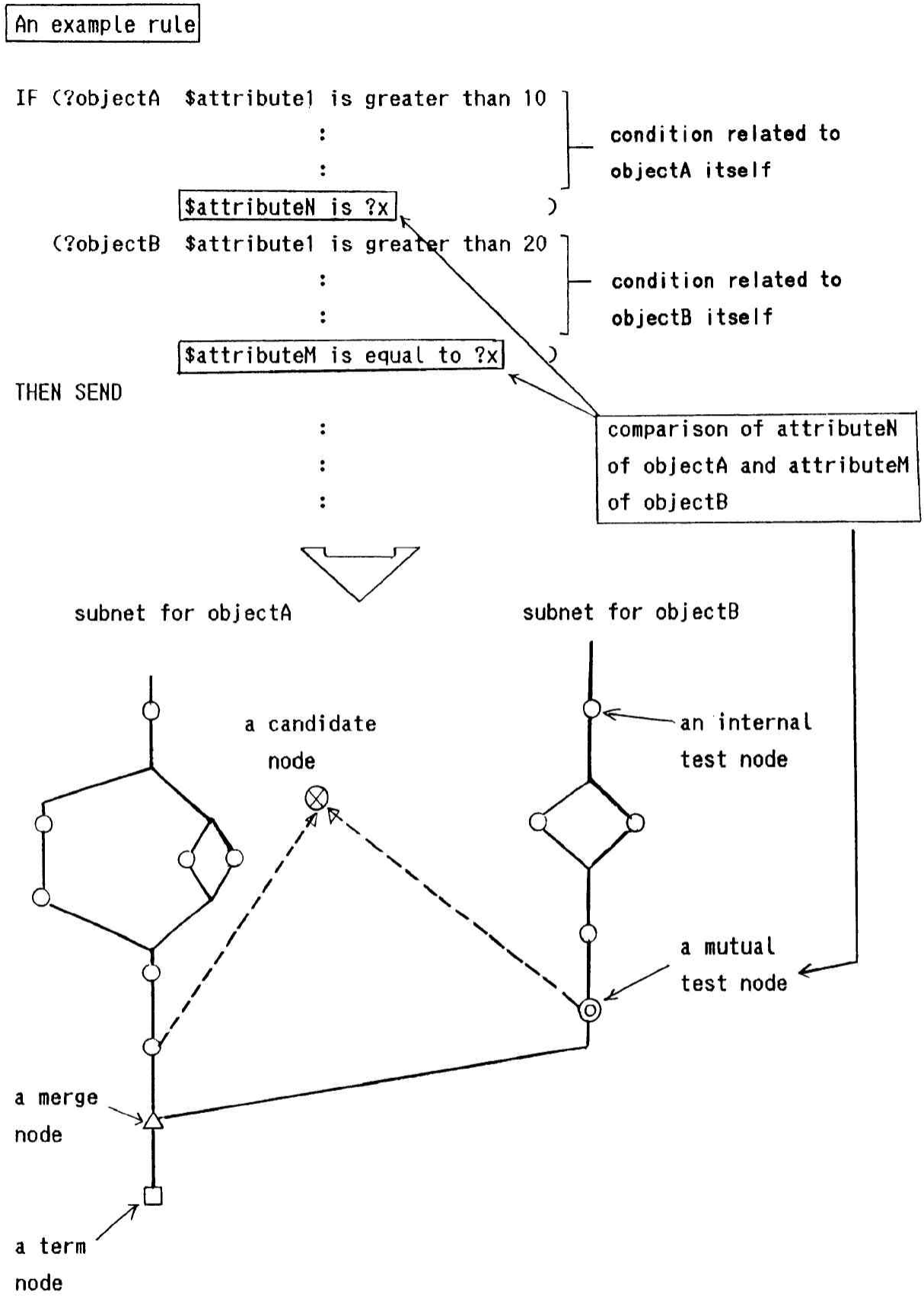


Figure 4 An Example Network for Condition Matching