ICOT Technical Report: TR-541

TR-541

A Proposal Guided Knowledge Acquisition Support System

by

H. Taki & S. Terasaki

April, 1990

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

A Proposal Guided Knowledge Acquisition Support System

Hirokazu Taki and Satoshi Terasaki

ICOT Research Center, Institute for New Generation Computer Technology,
4-28, Mita 1-Chome, Minato-ku, Tokyo 108 Japan
Csnet and Junet: htaki@icot.jp, terasaki@icot.jp

Abstract

This paper describes a proposal guided knowledge acquisition support system which generates and proposes candidates of knowledge by induction. This system is an enhanced version of the interactive knowledge acquisition support system EPSILON/One. This system elicits basic information from an expert, makes hypothesis, and modifies target knowledge with the expert. Some of the major problems of using induction is that it requires a large amount of computing power and creates numerous hypotheses. In order to make hypotheses by induction more efficient, the forms of hypotheses must be limited. EPSILON/One has a knowledge representation, Expert Model, which consists of operations. The operations are limited to seven types which are derived from analyzing real diagnostic knowledge bases. The system realizes efficient inductive hypothesis generation by using this limited numbers of operation representations. This papers discusses inductive operation presumption algorithms and interactions between the expert and the system in this architecture.

1. Introduction

A knowledge acquisition support system builds a knowledge base by interviewing an expert(Mizoguchi, R., Taki, H. and et al., 1988). The knowledge base must contain general knowledge to be applied to various cases(Kobayashi, S., Sawamoto, J., and et al., 1988).. The knowledge acquisition support system elicits target domain general knowledge directly from the expert by interview(Kawaguchi, A., Mizoguchi, R. and et al., 1978), while a learning system generates general knowledge from examples. This paper describes a knowledge acquisition system that has not only an interactive knowledge acquisition function that elicits general knowledge from the human expert, but also an induction function which presumes general knowledge from examples. Both the knowledge acquisition support system and the induction system have advantages and disadvantages. The induction system generates general knowledge from examples, but there are two disadvantages in generating this knowledge. It creates numerous candidate hypotheses of knowledge and requires a large amount of computing power. To reduce hypotheses and calculative complexity, the target form/language generated by induction must be limited. If the expert is not aware of general knowledge, there is a problem occurs where the interactive knowledge acquisition cannot extract enough general knowledge. We have evaluated our interactive knowledge acquisition system, EPSILON/One(Taki, H., 1989a)., and found that it must have presumed candidates of knowledge from example in some knowledge acquisition cases. The Expert Model(Taki, H. and Tsubaki, K., 1990), the knowledge representation of EPSILON/One, consists of seven types of operations which represent problem solving knowledge. These operation types limit the form of knowledge to be acquired. We also tried to use the special feature of the Expert Model to limit the form of consequence of induction. The knowledge acquisition support system with the induction function has a function for making proposals as general knowledge. We call this system the proposal guided knowledge acquisition support system. This paper introduces operation (knowledge) presumption and the proposal guided knowledge acquisition support system architecture.

2. Knowledge Acquisition and Induction

2.1 Knowledge Acquisition by Interview and Problems

The basic functions of the knowledge acquisition support system are knowledge elicitation, knowledge arrangement, knowledge refinement, and knowledge representation translation. A conventional approach to knowledge elicitation is through a psychological method to elicit general knowledge from the expert. The normal knowledge elicitation method stimulates the expert to remember new knowledge by redisplaying raw information or reformed knowledge which has been extracted from the expert. The personal construct psychology(Kelly, G.A., 1955; Boose, J.H., 1986) and the pre-post method(Taki, H., Tsubaki, K. and Iwashita, Y., 1987) which are supported by EPSILON/One are two examples of psychological methods. It is difficult for the expert to represent knowledge when he doesn't arrange and recognize knowledge related to problem solving(Tsubaki, K., Taki, H. and Osaki, H., 1989). However, when a kind of knowledge is proposed, he can judge easily whether it is used by him or not. Therefore, in order to enhance the knowledge elicitation function of the knowledge acquisition support system, it should propose candidates of knowledge for the expert. An example of a function which proposes candidates of knowledge is a function which can generate hypotheses that are more general than raw information extracted from the expert. Abduction, analogy, and induction are well known inference methods for generating hypotheses.

2.2 Integration of Knowledge Acquisition and Learning

Both the knowledge acquisition support system and the learning system are used for knowledge base building(Boose, J. and Gaines, B. (Eds.), 1989a; 1989b). However, they each have different approaches to building knowledge bases. The former acquires general knowledge from the expert directly. The latter generates hypotheses as knowledge from given information by advanced inference. A method of knowledge generation is a concept formation by inductive generalization. It creates general knowledge from examples. There are some problems in using induction for generating knowledge. It generates numerous hypothesis candidates and its calculation cost is high. For efficient induction, a language (form) of hypothesis representation must be limited to reduce candidates of knowledge, and domain specific knowledge must be used in the induction process to eliminate hypotheses(Genesereth, M. and Nilsson, N., 1986). We call the latter learning heuristics. A knowledge representation of problem solving is limited for efficient knowledge elicitation and refinement knowledge in the interactive knowledge acquisition. This limitation can also be used for language limitation of hypothesis generation by induction. Learning heuristics are dependent on the target domain. Therefore, the expert must cooperate with the induction system to reduce hypotheses by using heuristics. The interactive knowledge acquisition method is suitable for cooperative jobs with the expert. We explained that the knowledge acquisition support system and the learning system each compliment one another. A knowledge acquisition support system which is integrated into the learning system, can build a knowledge base by interviewing general knowledge and examples.

2.3 Knowledge Elicitation and Proposal

Basic functions of the knowledge acquisition support system with inductive functions are example elicitation, proposal generation by induction, and interactive editing proposal knowledge. The expert can require the next hypothesis, check modified knowledge using an inference engine, and add new example. These interactions between the proposal guided knowledge acquisition support system and the expert are shown in Fig.1.

3. Knowledge Acquisition Using Expert Model and Induction

3.1 Expert Model and Knowledge Acquisition

The knowledge representation of EPSILON/One is the Expert Model(Taki, H. and Tsubaki, K., 1990). It consists of seven types of operations. They are "classification", "selection", "input", "output", "ordering", "combination", and "translation". These types are derived from analyzed rules of knowledge bases in production rule form for diagnosis. The structure of the expert model consists of operations, a script which contains operation relations and inference control knowledge, evaluators which are the contents of operations, and elements which are evaluated in operations, as shown in Fig. 2. EPSILON/One supports a knowledge elicitation method (pre-post method) using the expert model. It consists of three stages; (1) operations and script elicitation, (2) evaluator elicitation using operation types and (3) element elicitation. In the second stage, knowledge is extracted effectively using operation type limitation. However, the knowledge to be extracted must be general knowledge. Therefore, if the expert is not aware of general evaluation criterion of an operation, the knowledge becomes very difficult to extract. However, it is not difficult for the expert to show examples which are input and output elements of the operation. The knowledge acquisition support system must contain a function which presumes general evaluation criterion from input and output element examples of the operation.

3.2 Inductive Knowledge Presumption Algorithm Using Operation Types

Operation presumption algorithms which uses operation types for consequence limitation of induction are introduced in this section. The seven operation types are "classification", "selection", "input", "output", "ordering", "combination" and "translation". The translation operation consists of three sub operations; "attribute value replacement operation", "mathematical calculation operation" and "element decomposition operation". To use induction, we must prepare "language which represents a target concept", "positive and negative examples related to the target concept", "knowledge which defines generalization of the concept", and "induction algorithms".

"Language which represents a target concept":

Evaluation criterion for each operation type

A basic criterion is related to one attribute.

"Positive and negative example related to the target concept":

Input and output elements of an operation

"Knowledge which defines generalization of the concept":

Generalization method for each evaluation criterion

"Induction algorithms"

Operation presumption algorithms for each evaluation criterion

EPSILON/One supports an efficient operation presumption function using operation types to limit language of inference consequence, treatment of examples, and methods of criterion generalization(Taki, H. and Fujii, Y., 1989). Here, we introduce three operation presumption algorithms; "selection and classification operation", "attribute value replacement operation" and "ordering operation". Currently, the algorithm of the combination operation is fixed and doesn't require any criterion. Solving the evaluator of the mathematical calculation operation is equivalent to solving simultaneous equations. Therefore, we don't prepare induction algorithms for combination operations and calculation operations. It is not necessary to presume evaluation criterion of "input operation", "output operation", and "element decomposition operation" because criteria of these operations are easily extracted from the expert. We explain the following items for three operation presumption algorithms:

(1) Evaluation algorithms of operations
(2) Evaluation criterion for each evaluation algorithm
(3) Treatment of examples
(4) Evaluation criterion presumption algorithms

3.3 Selection and Classification Operation Presumption

The selection operation selects elements which are suitable for a selection criterion from a given set. Classification operation classifies elements of a given set into sub sets according to classification criteria. This operation can be realized to combine several selection operations.

(1) Selection Algorithm

If an attribute and its value of an element satisfy a selection criterion, then the element is selected. The selection criterion consists of an attribute and its possible value sets or a conjunction of attributes and their possible value sets. If an element satisfies the selection criterion, the value of the attribute of the element is included in a value set of the criterion.

(2) Selection Criterion

A selection criterion related to one attribute is defined as follows:
(attribute-name, a set of possible values)

(3) Treatment of Examples

Examples are given as multiple elements of an input element group and an output element group. Elements selected by a selection operation are positive examples which represent a concept of selection criterion. Elements which aren't selected by the selection operation are negative examples. These examples (elements) are defined by the expert. Elements for examples have some attributes which have one value for each attribute.

Example-1:

Given an input and an output element group as follows.

Input element group = {e1, e2, ... , ei, ei+1, ... , en}

Output element group = {e1, e2, ... , ei} , ej (j = 1~n) means an element.

then

Positive examples = {e1, e2, ... , ei}

Negative examples = {ei+1, ... , en}

(4) Selection Criteria Presumption Algorithm

Step1: Making a set of attributes which are included in all examples.
This set is called "a total attribute list".

Step2: Making sets of values of each attribute in the total attribute list from positive examples. If an element doesn't have an attribute, its value is "undefined".

Step3: Making sets of values of each attribute in the attribute set from negative examples.

A matrix of the total attribute list (aj, j = 1˜m) and elements (ek, k = 1˜n) with values vjk (k = 1˜n, j = 1˜m), is made from the results of step2 and step3.

		a1	a2	a3 ...	am
Positive example	e1 :	v11	v21	v31 ...	vm1
	e2 :	v12	v22	v32 ...	vm2
			:		
	ei :	v1i	v2i	v3i ...	vmi
Negative example	ei+1 :	v1i+1	v2i+1	v3i+1 ...	vmi+1
	ei+2 :	v1i+2	v2i+2	v3i+2 ...	vmi+2
			:		
	en :	v1n	v2n	v3n ...	vmn

Sets of values for an attribute "aa" are defined as P(aa) when made from positive examples and N(aa) when made from negative examples.

P(a1) = {v11, v12, ..., v1i}
P(a2) = {v21, v22, ..., v2i}
:
P(am) = {vm1, vm2, ..., vmi}

N(a1) = {v1i+1, v1i+2, ..., v1n}
N(a2) = {v2i+1, v2i+2, ..., v2n}
:
N(am) = {vmi+1, vmi+2, ..., vmn}

Step4: Checking intersection of P(aa) and N(aa).

if $P(aj) \wedge N(aj) = \Phi$ (empty set) then select attribute aj
else reject attribute aj , (j = 1˜m)

If selected attributes are "aa", "ab" and "ac", P(aa), P(ab) or P(ac) is a selection criterion. Caution: As $P(aj) \wedge N(aj) = \Phi$ (empty set), $\neg N(aa) \wedge \neg N(ab) \wedge \neg N(ac)$ is also a selection criterion. $\neg N(aa)$ means that the value of the attribute "aa" isn't included in N(aa).

Step5: Showing selection criterion. If there are multiple criteria,
the system shows them in order.
In this example, the system shows P(aa), P(ab), P(ac), and $\neg N(aa) \wedge \neg N(ab) \wedge \neg N(ac)$ in this order.

(5) Guidance of generalization of selection criterion

The system shows how to make wider selection criterion by a set of attribute value generalization.

Example-2: { 1, 2, 3, 4, 5 } --> { >= 1, <= 5 } --> Integer
{ small-apple, big-apple } --> { apple } --> { fruit }

The system shows existence of elements which don't belong to either positive or negative examples.

Example-3: P(aa) = { 1, 2, 3 }, N(aa) = { 5, 6, 7 }

An element, where the value of an attribute "aa" is 4, doesn't belong to either P(aa) or N(aa). If a selection criterion is P(aa), this element is recognized as a negative example. If a selection criterion is $\neg N(aa)$, this element is recognized as a positive example. Therefore, $\neg N(aa)$ is a more general selection criterion than P(aa).

If there is no attribute aj which satisfies at $P(aj) \wedge N(aj) = \Phi$, then there is some dependency among attributes. In this case, the system must make selection criteria containing multiple attributes. (cf section 3.6).

3.4 Ordering Operation Presumption

(1) Ordering Algorithm

The ordering operation sorts elements according to ordering criterion related to one attribute. This ordering criterion is a total order. An order of elements means an ordering of processing elements in an operation.

(2) Ordering Criterion

An ordering criterion of one attribute is defined as follows:
(attribute-name, order list of values), (attribute-name, ascending),
or (attribute-name, descending)

(3) Treatment of examples

Examples are elements of an input element group and an output element group. An order of elements in the output element group is an ordering criterion of positive examples. Negative examples are relations of two items which differ from the input element group and the output element group. The input element group is not always necessary. Because there is no intersection between positive and negative examples, and negative examples aren't useful in limiting a criterion made from positive examples.

Example-4:
Given an input and an output element group as follows.

Input element group = {e1, e2, e3, e4}

Output element group = {e1, e3, e4, e2}

A relation that the expert prefers "a" to "b" is shown as "a >> b".
In this example, positive ordering relations are "e1 >> e3 >> e4 >> e2", negative ordering relations are "e2 >> e3" and "e2 >> e4".

(4) Ordering Criterion Presumption Algorithm

This algorithm deals with only the output element group as positive examples.

Step1: Making a set of attributes called a total attribute list which are included in positive examples.
Step2: Selecting one attribute from the total attribute list
Step3: Making a value list of the attribute according to the order of elements.
Step4: Checking inconsistent order pairs of two items.
An inconsistent order pair means that both "a >> b" and "b >> a" exist.
Step5: If there is no consistent relation, return to step2.
Step6: If elements of the value list are all numbers, then check whether their relation is ascending or descending. Even if they are numbers and the value list has other relations (ex. preference of odd number), the system treats them as literal symbols.

The system shows one ordering criterion . If there are some candidates of criteria related to different attributes, the system shows them in order.

3.5 Attribute Value Replacement Operation Presumption

(1) Attribute Value Replacement Algorithm

Values of a certain attribute of elements are replaced with different values according to a replacement criterion. This criterion is a replacement rule table which contains pairs consisting of a source value and a destination value.

(2) Replacement Criterion of Attribute Values

A replacement criterion of one attribute is defined as follows:

(attribuite-name, a set of (source value, destination value))

(3) Treatment of Examples

Examples are elements of an input and an output element group. Concerning an attribute, positive examples have different input and output values, negative examples have the same input and output values.

(4) Replacement Criterion of Attribute Values Presumption Algorithm

Step1: Searching for an attribute which has different values in elements of input and output. This attribute is called a remarkable attribute.

Step2: Making pairs of source values and destination values of the remarkable attribute in positive examples

Step3: Selecting combinations where the source value is the same but destination values are different. These are uncertain replacement rules.

Step4: Picking up values of the remarkable attribute in negative examples.

Step5: Checking whether the pairs created in Step2 have source values included in values made in Step4.
If a source value of a pair is included in values made in Step4, the pair is an inconsistent rule.

Step6: Displaying pairs which are not uncertain and inconsistent.

Step7: Displaying pairs which are uncertain and inconsistent for reference.

3.6 Evaluation Criteria Related to Multiple Attributes

We introduced basic presumption algorithms which treat criteria related to one attribute. These criteria are positioned in most general parts of a version spaces(Mitchell, T.M., 1978) shown in Fig.3. This section describes selection criteria related to multiple attributes. A method which we introduce in this section presumes complex criteria while it spreads target parts of the version space to include special concepts step by step. This method also uses the algorithm descried in section 3.2. In order to control the spreading the target part of the version space, this system uses learning heuristics from the expert through interaction. Selection criteria related to multiple attributes is represented in conjunction with some selection criteria of one attribute as follows:

(attribute-name-1, a set of values)∩(attribute-name-2, a set of values)∩...

Caution: The interview system of EPSILON/One elicits not only conjunctive form, but also disjunctive form and inequalities.

The version space (version graph) of the example in section 3.2 is shown in Fig.3. In this figure, the algorithm of section 3.2 processes (a)-part (P(aa), P(ab), P(ac), $\neg$N(aa)$\wedge\neg$N(ab)$\wedge\neg$N(ac)) of this figure.

To treat selection criteria related to multiple attributes, the system must deal with (b)-part in this figure. In order to stimulate the expert to remember necessary attributes, the system calculates similarities between positive and negative examples. This information enables the expert to remember which attributes are important for separating positive and negative examples.

Example-5:

Positive example:	at1	at2	at3	at4
e1	1	1	1	0
e2	1	1	0	1
e3	1	1	0	0
Negative example:				
e4	0	1	0	1
e5	1	0	0	0

In this example, positive and negative examples cannot be separated by a selection criterion related to one attribute. Therefore, the system calculates distances of pairs between positive and negative examples for all attributes.

distance(e1, e4) = | v11 - v41 | + | v12 - v42 | + | v13 - v43 | + | v14 - v44 |

If "a" is equal to "b", then | a - b | means 1 else it means 0.

Nearest pairs of a positive and a negative example are (e2, e5) and (e3, e5) which distances are 1. The system shows these pairs to the expert to order attributes according to his preference. The expert can eliminates unnecessary attributes. If the expert selects at2, at1 and at4 in this order, the ordering of hypothesis generation and testing is a sequence of "at2 $\wedge$ at1, at2 $\wedge$ at4, at1 $\wedge$ at4, at2 $\wedge$ at1 $\wedge$ at4". According to conjunctions of attributes, lists of attribute values are made.

The algorithm in section 3.2 processes one attribute as a conjunction of attributes and one value as their value list. For example, in the case related to two attributes, combination information of element "e1" is shown as follows:

pairs of attributes	(at1, at2)	(at1, at4)	(at2, at4)
pairs of values	(1, 1)	(1, 0)	(1, 0)

A candidate of selection criteria is (at1, at2) = { (1, 1) } in this example.

4. Proposal Guided Knowledge Acquisition Support System

4.1 Knowledge Acquisition by Proposal Generation

The basic frame work of proposal guided knowledge acquisition is to generate hypotheses from given information and to feedback these to the expert to remember new knowledge. Information of this interaction (cf. Fig.1) is shown as follows:

Information from the expert to the system:
(1) Examples (elements of input and output for an operation)
The two types of supply examples which are incremental supply and total supply.
(2) Knowledge modified by the expert (The expert can modify knowledge as hypothesis proposed by the system.)

Information generated by the knowledge acquisition support system
(3) Speciality in examples (ex. similarity of examples)
(4) General knowledge generated from examples by induction
(5) Results of evaluation of knowledge modified by the expert
The system evaluates modified knowledge with given input elements and shows output elements of the operation.

Requirements of the expert for the system
(6) Requirement of next hypothesis
(7) Requirement of evaluation of knowledge modified by the expert

Current implementation of the proposal guided knowledge acquisition support system is EPSILON/One and a sub-module which has an operation presumption function (shown in Fig.4). The target problem of knowledge acquisition is the diagnosis problem because the system handles seven operation types for diagnosis. This sub-module is implemented on PIMOS which is a parallel operating system on the PSI-2 (personal sequential machine) in a parallel logic programming language KL-1(Chikayama, K., and et al., 1989).

4.2 An Example

We introduce a simple example of the operation presumption about bearing selection, bearing ordering and their value replacement in a machine tool design problem(Inoue, K., Nagai, Y., and et al., 1988). In this example, the expert selects some bearing types from a total of twenty bearings(Nippon Seiko Corp., 1986), orders the bearings and replaces values of an attribute (cf. Fig.4).

(1)Presumption of Bearing Selection Knowledge

The expert sets twenty bearing types in an input element group and five bearing types in an output element group. These five bearings are positive examples and the rest of the bearings are negative examples. A bearing type consists of the following:

[deep-slot-ball-bearing, [force-of-radial, good],
[force-of-axial, good],
[direction-of-axial, bi-direction],
[mixed-force, good],
[high-speed, best],
[high-accuracy, best],
[low-noise, best],
[possible-sloop, better],
[fixed-case, possible],
[free-case, conditional]]

Examples are given as follows:
Input element group = [twenty types of bearings]
Each bearing has an attribute "high-accuracy" which values are
"best", "better", and "undefined".
Output element group = [deep-slot-ball-bearing, angular-ball-bearing, cylinder-roll-bearing, combinatorial-angular-ball-bearing, multiple-cylinder-roll-bearing]
A presumption result is high-accuracy = [[best]](shown in Fig.6 in Japanese). Fig.6 also shows some windows of criteria related to two attributes and three attributes.

(2) Presumption of Bearing Ordering Knowledge

The expert sets an ordering of the five bearings in an output element group. The ordering is as follows:

Input element group = [
[deep-slot-ball-bearing, [high-speed, best], ..],
[angular-ball-bearing, [high-speed, best], ..],
[cylinder-roll-bearing, [[high-speed, best], ..],
[combinatorial-angular-ball-bearing, [[high-speed, better], ..],
[multiple-cylinder-roll-bearing , [[high-speed, better], ..]]

This ordering is related to the attribute "high-speed", so the system shows
"high-speed = [best, better]" as a presumption result.

(3) Presumption of Bearing Attribute Value Replacement Knowledge

The expert sets an output element group. The input element group of this operation is equal to the output element group of the ordering operation. A value of an attribute of each element in the output group is different from one in the input element group.

Examples are shown as follows:

Output element group = [
[deep-slot-ball-bearing, [high-speed, range-4], ..],
[angular-ball-bearing, [high-speed, range-4], ..],
[cylinder-roll-bearing, [[high-speed, range-4], ..],
[combinatorial-angular-ball-bearing, [[high-speed, range-2], ..],
[multiple-cylinder-roll-bearing , [[high-speed, range-2], ..]]

This replacement is related to the attribute "high-speed", so the system shows
"high-speed = [[best -> range-4], [better -> range-2]]" as a presumption result.

5. Related Works

Case based reasoning(Kolodner, J.L., Simpson, R.L. and Sycara, K., 1985) is a method which searches for a similar example from a case base through analogy and modifies it to suit a target problem. The knowledge acquisition support system prepares a knowledge base which contains general knowledge before reasoning, while the case base reasoning uses the case base directly during reasoning by generalization and analogy.

ID3(Quinlan, J.R., 1983) is an induction algorithm for generating a classification tree. Its classification criterion is an efficient search of the classification tree. Our operation presumption algorithm solves a classification criterion from given leaves of the classification tree, while ID3 generates the classification tree and leaves of the tree from a given classification criterion and items to be classified.

The MIS (Model Inference System)(Shapiro, E.Y., 1981)solves a horn clause by induction. It specifies a horn clause with the oracle by asking yes/no questions. Candidates of the horn clause are defined in the limited space of refinement operators. This search space is a kind of version spaces. The MIS prunes unnecessary candidates efficiently through skillful questions. Our operation presumption algorithm deals with more special forms of knowledge and more complex interaction than the MIS.

AQUINAS(Boose, J.H. and Bradshaw, J.M., 1987) is a knowledge acquisition support system for classification problems. It extracts classification knowledge in the form of a rating grid. It also makes an implication graph which includes many hypotheses of implication relations among traits. The expert can refine these implications. This refinement style is a kind of proposal guided knowledge acquisition. The induction method of AQUINAS is simpler than our method.

6. Current Problems

In the case that the expert cannot supply learning heuristics, the complexity of current operation presumption algorithm for selection criteria requires a large amount of computing power and is equal to the complexity of a total solution search of a version space. Therefore, it is important to use an elicitation method of learning heuristics. For this problem, the speciality of examples must be illustrated with a cluster tree to represent similarities between examples.

7. Summary

We introduced a knowledge acquisition support system which makes hypotheses by induction and proposes candidates of knowledge. This research suggests a suitable integration of the engineering knowledge acquisition method and logical learning method. An important point is usage of operation types to limit forms of consequence of induction. This system has been implemented in parallel logic programming language GHC (KL1) on a PSI-2 inference machine developed by ICOT. We are planning to apply this system to not only diagnosis problems(Nagai, Y., Taki, H., and et al., 1989) but also to design problems which are a kind of constraint problem solving problems(Terasaki, S., Nagai, Y., Yokoyama, T., Inoue, K,, Horiuchi, E. and Taki, H., 1988) .

Acknowledgement

We wish to express our thanks to members and the chief of ICOT 5th Laboratory, Mr. Kenji Ikoma, and members of the PGF and KSA working groups. We would also like to thank Mr. Marc Linster at GMD for his good suggestions. We must express our thanks to the Director of ICOT research Center, Dr. Kazuhiro Fuchi.

References

Boose, J.H. (1986). A Knowledge Acquisition Program for Expert Systems Based on Personal Construct Psychology, J. of Man-Machine Studies, Vol. 23, pp 495-525

Boose, J.H. and Bradshaw, J.M. (1987). Expertise Transfer and Complex Problem Using AQUINAS as A Knowledge-Acquisition Workbench for Knowlegde-Based Systems, J. of Man-Machine Studies, Vol.26, No.1, pp3-28

Boose, J. and Gaines, B. (Eds.) (1989a). A SUMMARY OF THE THIRD EUROPEAN WORKSHOP ON KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS, Knowledge Acquisition for Knowledge-based Systems Newsletter, Vol.1, No.2, Aug.

Boose, J. and Gaines, B. (Eds.) (1989b). A SUMMARY OF THE FOURTH KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP, Knowledge Acquisition for Knowledge-based Systems Newsletter, Vol.1, No.3, Nov.

Chikayama, K., and et al. (1989). Overview of Parallel Inference Machine Operating System(PIMOS), ICOT TR No. 483

Genesereth, M. and Nilsson, N. (1986). Logical Foundations of Artificial Intelligence, Los Altos, CA: Morgan-Kaufman

Inoue, K., Nagai, Y., Fujii, Y., Imamura, S. and Kojima, T. (1988). Analysis of Design Method of Machine Tools,ICOT Technical Memorandum, No.494 (in Japanese)

Kawaguchi, A., Mizoguchi, R., Yamaguchi, T. and O. Kakusho, O. (1978). SIS: A Shell for Interview Systems, Proceeding of IJICAI-79, Vol.1, pp359-361

Kelly, G.A. (1955). The Psychology of Personal Constructs, New York, Norton

Kobayashi, S., Sawamoto, J., and et al. (1988). Knowledge Acquisition Support for Analytical Knowledge Systems, ICOT TM No.581 (in Japanese)

Kolodner, J.L., Simpson, R.L. and Sycara, K. (1985). A process model of case-based reasoning in problem-solving, Proceedings of IJICAI-85

Linster, M. (1989). Report on my Visit to ICOT in August

Mitchell, T.M. (1978). Version Spaces: An Approach to Concept Learning, Stanford Technical Report STAN-CS-78-711, HPP-79-2

Mizoguchi, R., Taki, H. and et al. (1988). Knowledge Acquisition Support Sytems, ICOT TM No. 654 (in Japanese)

Mizoguchi, R., Taki, H. and et al. (1989). An Image of Knowledge Acquisition Support Systems for Design Problems, ICOT TM No.747 (in Japanese)

Nagai, Y., Taki, H., Terasaki, S., Yokoyama, T. and Inoue, K. (1989) A Tool Architecture for Design Expert Systems, Journal of Japanese Society for Artificial Intelligence, Vol.4, No.3, pp61-67 (in Japanese)

Nippon Seiko Corp.(1986). Roll Bearing (Catalogue) (in Japanese)

Quinlan, J.R. (1983). Learning Efficient Classification Procedures and Their Application to Chess End Games, in Machine Learning: An Artificial Intelligence Approach, R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), Tioga, Palo Alto, Calif.

Shapiro, E.Y. (1981). Inductive Inference of Theories from Facts, Technical Report 192, Yale University Computer Science Dept.

Suwa, M., Sawamoto, J., and et al. (1988). Knowledge Acquisition Support for Synthesis Knowledge Systems, ICOT TM No. 588 (in Japanese)

Taki, H., Tsubaki, K. and Iwashita, Y. (1987). Expert Model for Knowledge Acquisition, Proceedings of Proceedings of IEEE Expert Systems in Government Conference

Taki, H. (1989a). A Knowledge Acquisition Support System EPSILON/One, Proceedings of 7th Symposium of Fifth Generation Computer, pp17-18 (in Japanese)

Taki, H. (1989b). Knowledge Acquisition by Abductive and Inductive Explanation, Proceedings of 4th Knowledge Acquisition for Knowledge-Based Systems Workshop, pp 34-1 - 34-19

Taki, H. and Fujii, Y. (1989). Operation Presumption: Knowledge Acquisition by Induction, Proceedings of Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems, pp 34-48

Taki, H. and Tsubaki, K. (1990). Expert Model: A Knowledge Representation for Knowledge Acquisition, Journal of Japanese Society for Artificial Intelligence, Vol.5, No.2 (in Japanese) (to appear)

Taki, H. (1990). Induction Programs for Operation Presumption in KL1 Language , ICOT Technical Memorandum (to appear)

Terasaki, S., Nagai, Y., Yokoyama, T., Inoue, K,, Horiuchi, E. and Taki, H. (1988). A Machine Design Support Systems Building Tool -MECHANICOT-, Journal of Japanese Society for Artificial Intelligence, SIG-KBS-8803 (in Japanese)

Tsubaki, K., Taki, H. and Osaki, H. (1989). A Knowledge Acquisition Support System EPSILON/One (a evaluation report), Proceedings of 3rd JSAI Conference, Vol.2, 12-5, pp 719-722 (in Japanese)

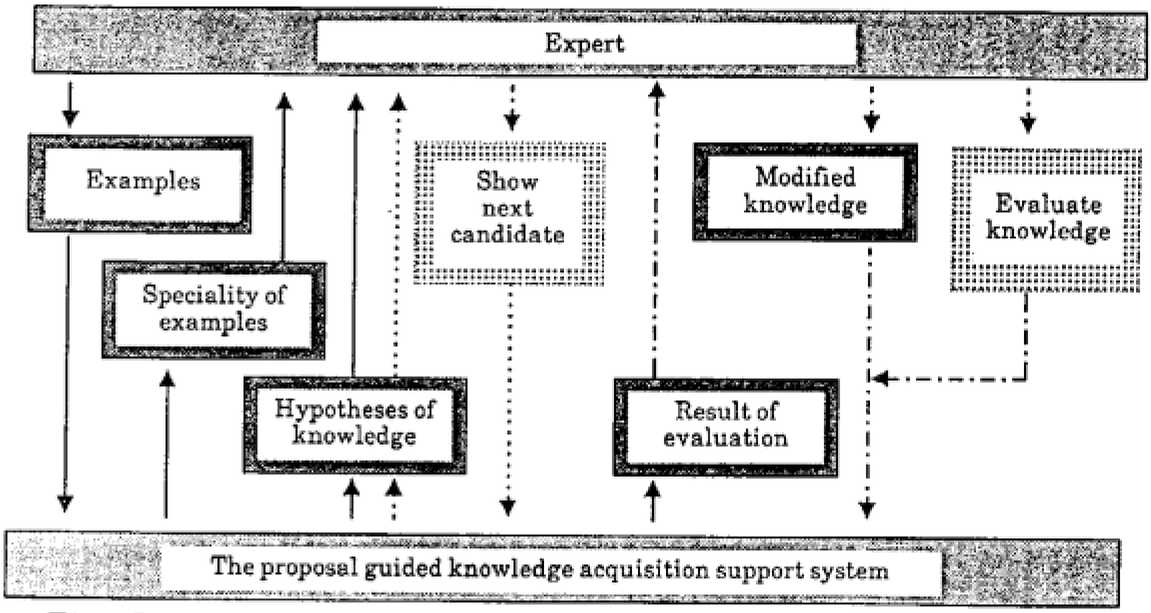


Fig.1 Interaction between the proposal guided knowledge acquisition support system and the expert

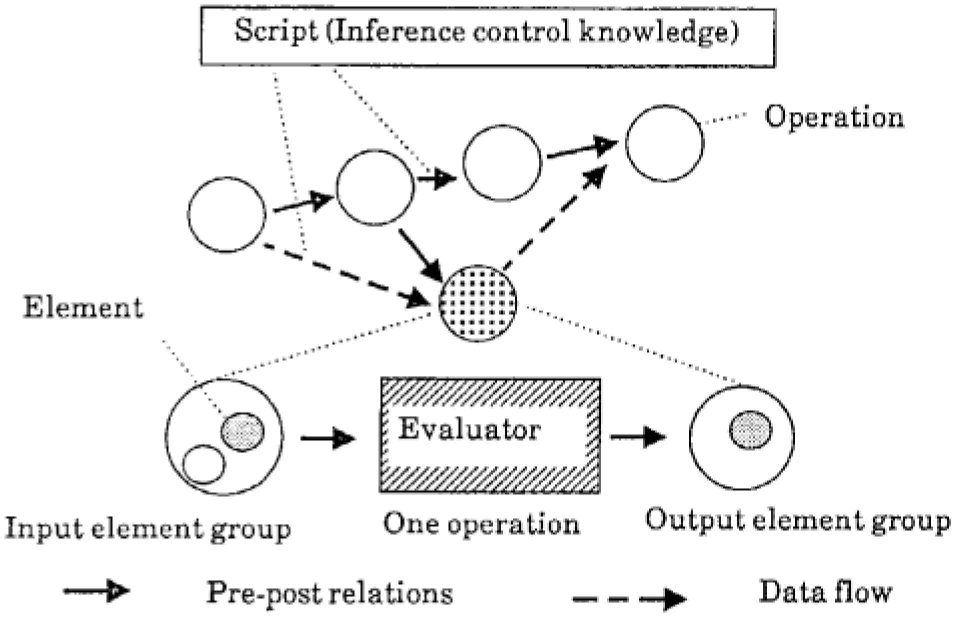
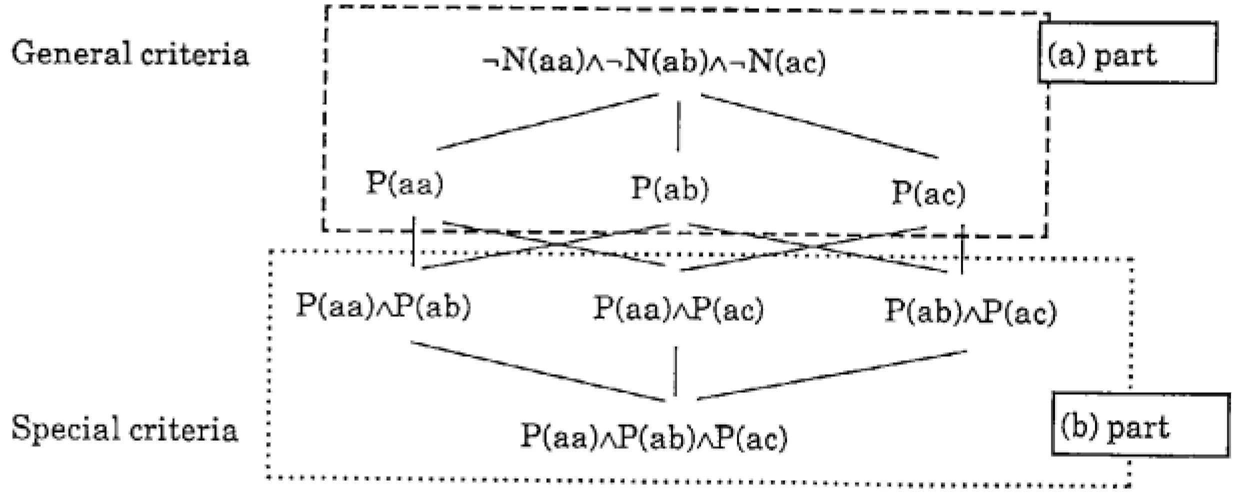


Fig.2 the Expert Model basic structure



P(aa)∧P(ab) means that the value of an attribute "aa" belongs to P(aa) and the value of an attribute "ab" belongs to P(ab).

Fig.3 A version space

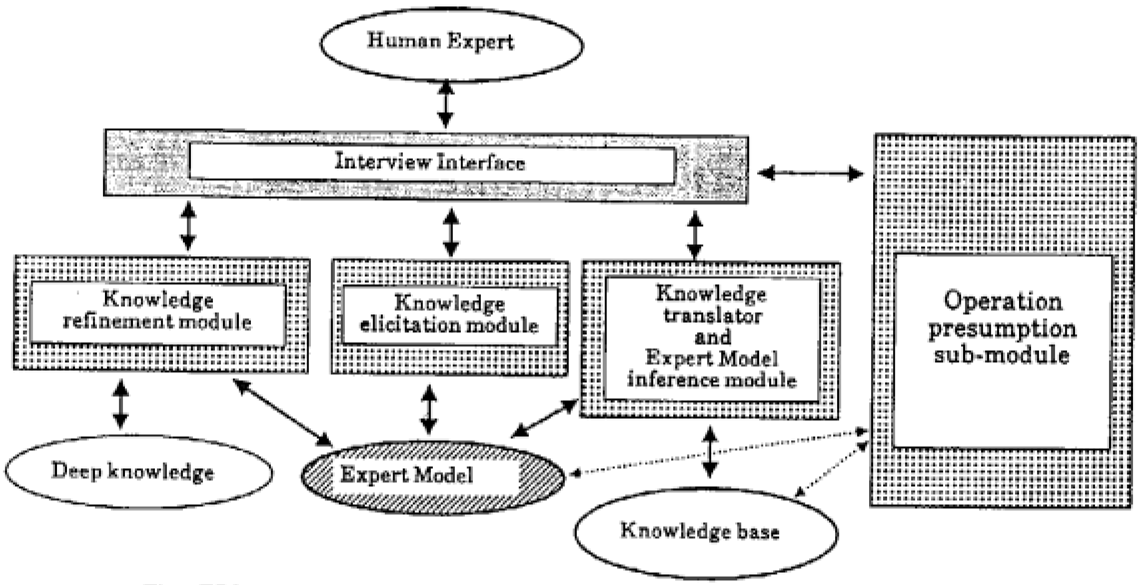


Fig.4 EPSILON/One and operation presumption sub-module

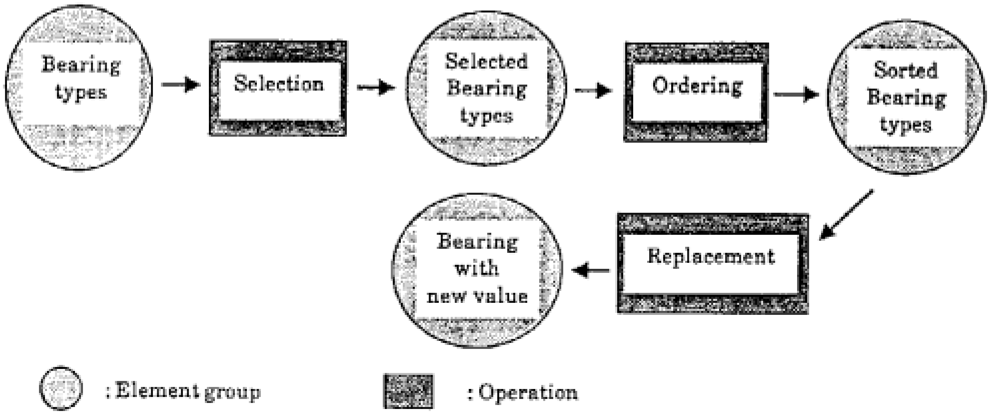


Fig.5 Example

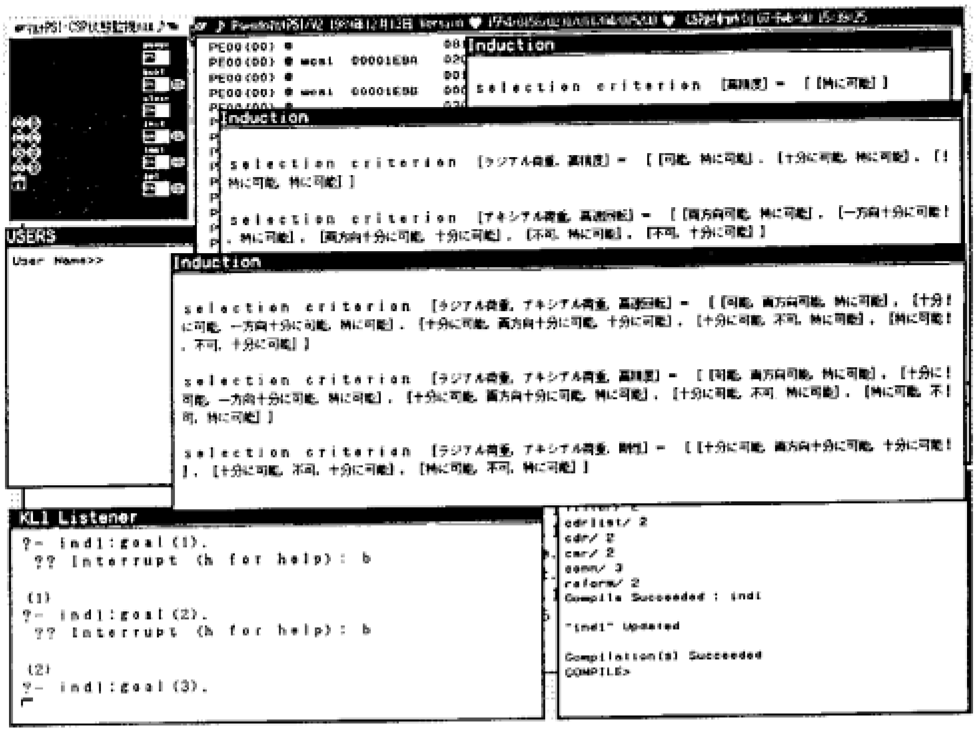


Fig.6 An example of Operation presumption on PSI-2