ICOT Technical Memorandum: TM-0725

TM-0725

Knowledge Acquisition by abductive and Inductive Explanation (Preliminary Report)

by
H. Taki

May, 1989

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

Knowledge Acquisition by Abductive and Inductive Explanation
(Preliminary Report)

Hirokazu Taki

ICOT Research Center, Institute for New Generation Computer Technology,
4-28, Mita 1-Chome, Minato-ku, Tokyo 108 Japan
Phone: +81-3-456-3192, Csnet:htaki%icot.jp@relay.cs.net

Abstract

This paper describes an explanation-based learning system under incomplete knowledge. We propose an explanation method using abduction and induction. If the system fails to explain a goal and to explain examples, it makes a hypothesis by abduction. In this way, there are many candidate hypotheses. The system uses a few criteria to eliminate candidates. Selection of hypotheses that are in a consistent explanation is explained by induction using a background theory and has minimal generation cost. After explanation generation, the explanation tree is generalized according to operationality criteria. As a result, it generates assumptive macro knowledge.

1. Introduction

Knowledge acquisition bottleneck is one of most difficult problems in the building of knowledge bases of expert systems. There are two ways to build knowledge bases. The first way is interactive knowledge acquisition: a human expert is interviewed to extract his knowledge. The other way is knowledge building by understanding observed information. One way to understand something is to try to explain it. EBL(Explanation-based learning)[Mitchell 85][Mitchell 86] is a learning method which uses explanation of examples. However, EBL makes only macro knowledge for effective inference; it doesn't make new knowledge that solves new problems. In order to use EBL, a complete domain theory must be prepared before learning. Normally, however, it is difficult to prepare complete knowledge before learning. Therefore, a learning system fails to explain examples. This situation is a trigger which makes other learning occur. Impasse situations of chunking in SOAR[Laird 86] and frustration situations of FBL[Suwa 89] are like failure situations of explanation. Abductive explanation is a kind of explanation method which uses hypotheses. In this explanation, a hypothesis is made by abduction[Charniak 86]. A hypothesis is new knowledge if it does not derive inconsistency. There is a shortcoming in abduction, which generates many candidates as hypotheses to explain examples. It is important to select meaningful hypotheses from all hypotheses by abduction. This paper introduces a hypothesis selection method which considers the operation cost of generalization and specialization of a domain theory. In this method, a hypothesis is selected syntactically using the following criteria.

(1) An explanation has no inconsistency.

(2) A hypothesis made by abduction is explained from concepts of version spaces[Mitchell 77] which were generated from facts and deductive conclusions of the domain theory.

(3) An explanation should be selected that has the lowest total cost of hypothesis generation

To consider the relation between generalization level and meaning of concept representation level (such as structure information of objects and attributes of objects) , this method can also deal with semantic hypothesis selection.

After abductive explanation generation, to generalize this explanation tree, an assumptive macro knowledge is obtained by EBL.

In ICOT, a knowledge acquisition system, EPSILON[Taki 87], has been developed as an interactive knowledge acquisition system which creates an initial knowledge base based on an expert model represented by primitive operations. EPSILON has been implemented on PSI(Personal Sequential Inference Machine). The learning method by abductive explanation is developed as a learning module of EPSILON which extracts an assumptive knowledge after initial knowledge base building.

2. Abductive Explanation

When we want to explain something but there is insufficient information, we make hypotheses. These hypotheses are either uncertain knowledge or assumed knowledge. The following sections describes how to assume knowledge for explanation. In the discussion, we assume that background theory, a goal and examples are given in learning process. This background theory is not enough to explain the goal with examples.

2.1 Explanation under Incomplete Knowledge

In knowledge acquisition, it is important to understand observed data and given examples. Understanding examples means to give an explanation as to why some examples are given to realize a goal. Usually we explain a physical behavior by stating a physical law. In this case, the physical law is background knowledge and we explain the behavior as a goal with physical situations as examples. If some physical situations or a few physical law are unknown, we cannot explain the physical phenomenon. However, we can make assumptions to try to explain that phenomenon.

(Example 1) Explanation about a broken bridge

When we find a broken bridge, we wonder how the bridge broke. We didn't see it happen, so we must guess, . We assume that bad weather broke the bridge.

An explanation: Rain fell.(assumption)
The river rose.
The bridge spans the river.
The swollen river was very strong.
The river pushed against the bridge.
The river was stronger than the bridge.
The bridge broke.

Assumed information is a hypothesis. There are two types of hypotheses: one of them is in a given hypothesis set which includes uncertain knowledge (we don't know whether the knowledge is true or false), another is generated by advanced reasoning (such as analogy, abduction or induction). The assumption of the weather in example 1 is an uncertain hypothesis. We know candidates for the weather; rain, wind, snow, fine and cloudy. We select a candidate from this finite hypothesis set. This gives us a macro knowledge: if it rains, a bridge may break. This process is like EBL without using hypotheses.

(Example 2) Explanation about a flying beetle

Tom knows that small insects can fly but he doesn't know that a beetle can fly. One day, a big flying insect crashed into Tom's head. Tom saw that the insect had a horn.Tom was surprised to see such a big insect fly. Tom assumes that any insect can fly.

An explanation: Tom knows that small insects can fly, so he induces that any insect can fly. (assumption)
Therefore, that flying object is a big insect.(assumption)
The big insect has a horn. So Tom identifies it as a flying beetle.

A hypothesis in example 2 is generated by inductive inference. Tom thought that a concept of "flying small insects" was generalized and a concept of "flying insects" was generated. He eliminated the size attribute of insects. This example shows that hypothesis generation is important to explain examples under incomplete knowledge and this hypothesis is generated only if it is necessary.

2.2 Explanation by Hypothetical Reasoning

In explanation with hypotheses, there may be inconsistency. Hypothetical reasoning deals with inconsistent reasoning during hypothesis selection. A formulation of a hypothetical reasoning[Pool 88] is as follows.

B: a background theory, H: a set of hypotheses, E: examples,
G: a goal, h: a subset of H

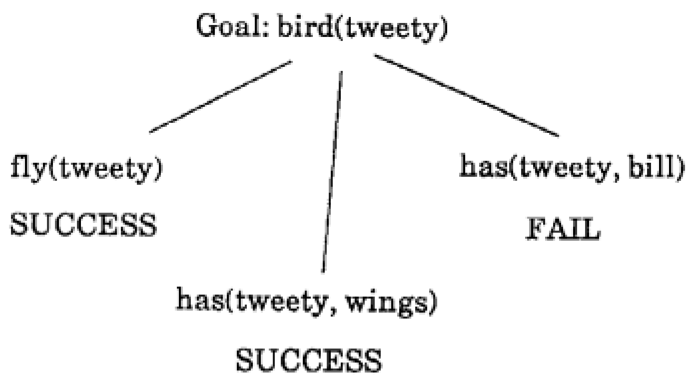
$h \in H$
$B \cup E \vdash G$
$B \cup E \cup h \vdash G$
$B \cup E \cup h \vdash \perp$

We eliminate the first formula and add the hypothesis generation formula.

$B \cup E \vdash G$
$B \cup E \cup G \mid< h$ (Induction)
$B \cup E \cup h \vdash G$
$B \cup E \cup h \vdash \perp$

This formula is shown in induction[Genesereth 86]. We use the latter formulation. We think that this formulation is suitable not only for induction but also for abduction and analogy.

Figure 1 shows an explanation system using ATMS[de Kleer 86] as hypothetical reasoning.



2.3 Generation of Hypothesis by Abduction

There are two kinds of candidates for hypotheses in this framework. They are fact from knowledge and rule form knowledge. We deal with the fact form knowledge as hypothesis. It is difficult to generate hypothesis deductively. This problem contains when, how and on what to generate hypotheses. When a reasoning process fails to make an explanation, this situation is a trigger to make a hypothesis. Abduction is the process that generates explanations. Abduction has the following form.

Given "if a then b"
Given "b"

Infer "a"

A prolog-like abduction form to use Horn clause is shown as follows.

Given b(X) :- a(X)
Given b(c)

Infer a(c)

Here is an example which makes an explanation by abduction.

(Example 3) Explanation about a bird, tweety
Background theory: bird(X) :- fly(X), has(X, wings), has(X, bill).
Examples: has(tweety, wings)
fly(tweety)
Goal: bird(tweety)

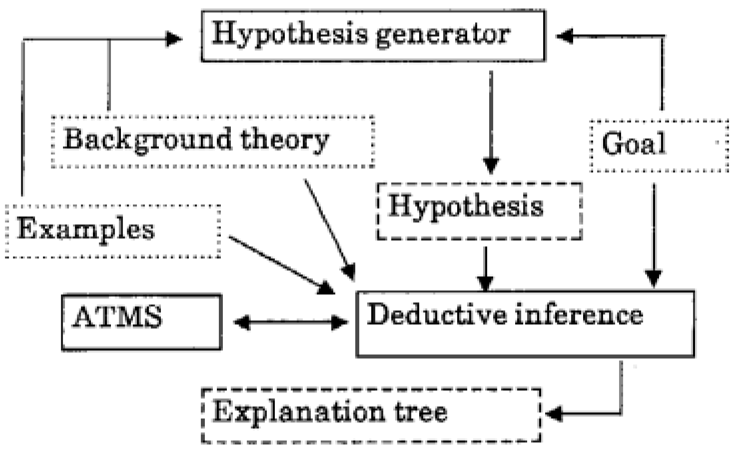


Figure 1 The explanation system

Given bird(X) :- fly(X), has(X, wings), has(X, bill)
Given bird(tweety)
------------------------ (ABDUCTION)
Infer has(tweety, wings), fly(tweety), has(tweety, bill)

"has(tweety, wings) and fly(tweety)" are known. Therefore a result of abduction is "has(tweety, bill)". If there is the hypothesis, "has(tweety, bill)", then the explanation of example 3 is established.

3. Selcction Abductive Explanation

If we use abduction for any explanation, we can explain anything. It is necessary to select good hypotheses in order to generate better explanations. Selection of hypotheses means selection of explanations. This section describes an evaluation as selection criteria.

3.1 Selection Hypothesis

We define the following criteria to select hypotheses.

(1) An explanation has no inconsistency.
(2) A hypothesis made by abduction is explained from concepts of version spaces which are generated from facts and deductive conclusions of the domain theory.
(3) The explanation with the lowest total cost of hypothesis generation should be selected

The second criterion selects hypotheses which are explained by induction. If there are two hypotheses in an explanation, the total cost is the sum of both hypothesis generation costs.

3.1.1 Types of Hypothesis

To begin with, types of hypothesis in the explanation system are introduced. Necessary hypothesis is dependent on a sequence of explanation. In example 3, if a goal concept is bird(X), there is a difference between rule 1 "bird(X) :- fly(X), has(X, wings), has(X, bill)" and rule 2 "bird(X) :- has(X, bill), fly(X), has(X, wings)".

In an inference using the rule 1, when "has(X, bill)" is checked, the value "X" has been unified to "tweety" which was fixed at checking "fly(X)". Therefore, a result of abduction in this case is "has(tweety, bill)". On the other hand, when "has(X, bill)" is checked in the case of the rule 2, the value "X" is not fixed. The result of abduction is "has(X, bill)". After reasoning, both explanations are the same. In the second case, as a result, "has(X, bill)" is "has(tweety, bill)". This shows that necessary hypothesis is "has(tweety, bill)" in both cases. We call this result of abduction "has(tweety, bill)" a basic hypothesis and the result "has(X, bill)" a temporal hypothesis.

If there are a basic hypothesis and a temporal hypothesis from an item by abduction in an explanation, the temporal hypothesis is more general than the basic hypothesis and the basic hypothesis is the most special hypothesis in the explanation.

3.1.2 Version Space and Hypothesis

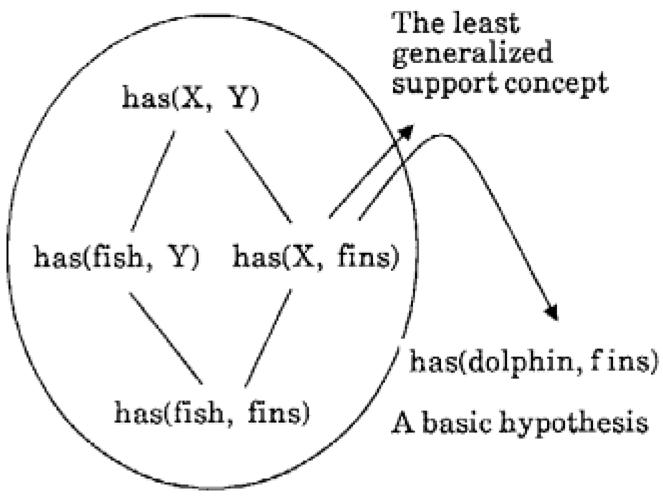
Knowledge in the background theory can explain only concepts that are more specific than itself. Generalized knowledge is necessary to explain a hypothesis. Sets of generalized knowledge are represented by version spaces made from the background theory.

Normally, the induction function decides an upper and lower boundary of a version space using positive and negative examples of a concept.

If the background theory doesn't include negative examples, version spaces of concepts in the background theory have no upper boundaries. Therefore, the most generalized form of an item has its own predicate name and arguments that are represented by valuables. This means that the predicate name is not generalized. The lower boundaries of those version spaces are defined by facts that are deduced from the background theory.

A concept in a version space which explains a basic hypothesis is called "a support concept for the basic hypothesis". The most specific concepts of support concepts is called "the least generalized support concept". The least generalized support concept is one of concepts made from the basic hypothesis and facts in the background theory by least generalization.

(Example 5) The least generalized support concept



Version space

In example 5, "has(fish, fins)" is the lower boundary of the version space, "has(X,Y)" is the upper boundary of it. "has(X, fins)" and "has(fish, Y)" are concepts in the version space. If "has(dolphin, fins)" is a basic hypothesis, "has(X,Y)" and "has(X, fins)" are support concepts for the hypothesis because these unify to it. A least generalized support concept is "has(X, fins)".

If "has(beetle, horn)" is a basic hypothesis, a least generalized support concept is "has(X, Y)". If "eat(dolphin, fish)" is a basic hypothesis, there is no support concept for this hypothesis in this version space.

3.1.3 Hypothesis Generation Cost

This section describes operations for predicate modification. Each operation has the same cost and is identified as 1 point. Three operations are defined here. The first operation, argument generalization operation (G-op), exchanges a fixed argument to a variable. The second operation, argument disconnect operation (D-op), eliminates a dependency between two variables. The third operation, argument unification operation (U-op), unifies a instance and a variable.

(Example 6) Operations

G-op: has(dolphin, fins) -> has(X, fins) or has(dolphin, Y)
D-op: has(X,X) -> has(X,Y)
U-op: has(X, fins) -> has(fish, fins)

In example 5, total cost of generation "has(dolphin, fins)" from "has(fish, fins)" is 2 points. This generation process consists of "G-op: has(fish, fins) -> has(X, fins)" and "U-op: has(X, fins) -> has(dolphin, fins)".

4. Assumptive Macro Knowledge Generation

In this section, this explanation function is integrated into EBL. We can obtain assumptive macro knowledge by abductive explanation-based learning.

4.1 Explanation-Based Learning and Incomplete Domain Theory

EBL is a learning about efficiency of knowledge. In EBL, complete domain theory, goal concept, learning examples and operationality criteria must be prepared. EBL explains a goal using examples and domain theory. An explanation tree is generalized by operationality criteria. EBL doesn't learn new knowledge. Before learning, it is hard to prepare a complete domain theory. If a framework of EBL under incomplete domain theory were developed, it could learn new knowledge and efficiency of knowledge usage. There is some incompleteness in the domain theory as follows.

(1) Completeness: There is some lack of knowledge, so some examples cannot be explained.

(2) Soundness: There is some wrong knowledge, so wrong examples are explained.
(3) Consistency: The domain theory contains inconsistency, so inconsistency is detected in an explanation.
(4) Over-generalization: Knowledge is too general, so it explains negative examples.
(5) Over-specialization: Knowledge is too special to explain positive examples.

We aim to develop this framework using abduction and induction. We aim to develop an explanation system which deals with incomplete knowledge (1) and (5). It makes assumptive explanations and generalizes explanations. As a result, it generates assumptive macro knowledge which contains hypothesis as a new knowledge and usage of the hypothesis.

4.2 Assumptive Macro Knowledge

A basic hypothesis is very special for learned knowledge. A least generalized support concept is a support hypothesis for a basic hypothesis. Therefore, we use the least generalized support concept as a hypothesis to learn new knowledge. If the support concept is denied, the basic hypothesis is also denied. The support concept may explain other hypotheses different from the basic hypothesis. However, examples and an explanation are special cases. The learning system must learn more general inconsistent knowledge. We define assumptive macro knowledge shown in a following form.

Knowledge generated by EBL:

goal(X1, ..., Xn) :- p1(Xi, .., Xj, ..), p2(...), ..., pm(...). --(1)

Hypothesis: h1(Xk, dolphin).
Assumptive macro knowledge (represented in default rule form):

$$\frac{\text{p1(...), p2(...), ..., pm(...) M: h1(Xk, dolphin)}}{\text{goal(X1, ..., Xn)}}$$

Unless "h1(Xk, dolphin)" is not denied, a horn clause (1) is available.

5. Semantic Hypothesis Selection

The assumptive macro knowledge is generated syntactically. This section shows how we can express semantics about generalization and specialization.

5.1 Generalization Level of Hypothesis

The least generalized support concept is generated from the background theory and examples by generalization. However, there are some limits for generalization. These limits decide the upper boundaries of version spaces made from known knowledge such as negative examples. These constraints are dependent on application fields. For example, structured mapping theory[Falkenhainer 87] controls analogical mapping between two events. It maps structure information but not attributes of objects in these events. In order to control generalization levels of special predicates, this system allows the user to limit version spaces of predicts which are defined in a set of special names or have a fixed number of arguments.

5.2 Operationality

In order to generate operational macro knowledge, EBL generalizes an explanation tree about its structure. In our learning method, there are two relations between a hypothesis and an operationality criterion.

(1) An operationality criterion is more special than a hypothesis.

In this case, the hypothesis exists in macro knowledge. If this macro is used in an inference, this hypothesis must be explained by deduction. This type of macro knowledge is normal knowledge but not default knowledge.

(2) An operationality criterion is more general than a hypothesis.

In this case, the hypothesis doesn't exist in macro knowledge. It is eliminated from the explanation tree by an operatinality criterion. Therefore, this macro is independent of knowledge whose generalization level represents the hypothesis. The hypothesis is no longer necessary to explain the macro knowledge. However, if the hypothesis is denied, the macro knowledge loses its generation cause/reason. This type of knowledge is represented in assumptive macro knowledge form.

6. Related Works

Our research is a kind of EBL under incomplete knowledge[Doyle 86][Rajamoney 88][Uchihashi 89]. A characteristic of this learning is to explain by abduction and induction. The research of abductive explanations in natural-language interpretation in SRI[Stickel 88] also discusses abductive explanation and assumption cost. Its explanation is based on abduction. We use abduction to make a basic hypothesis and decide least generalized support concept using a version space. Therefore, our explanation is based on induction. The integration of EBL, and ATMS or hypothetical reasoning [Atsumi 88] is developed for making consistent explanations. Our system also uses ATMS to check explanation consistency. A default rule maintenance system[Araragi 89] uses a trigger when an inconsistent explanation is found. It modifies a default representation of a default rule. Our system doesn't maintain default rules, but eliminates inconsistent explanations and macro knowledge. The learning method to generalize explanation trees with the integration method of similarity-based learning and operationality criteria is developed[Yamamura 87]. Our system uses induction to generate explanations but not to generate a generalized explanation tree from some explanation trees. A model inference system[Shapiro 81] uses a refinement tree to specialize predicates with six strategies. The current version of our system uses only three operations to generalize and specialize knowledge. Its next version supports the following operations.

List generalization operation 1: has(X,[Y|Z]) -> has(X,Y)
List generalization operation 2: has([X|Z],Y) -> has(X,Y)
Implication eliminate operation: has(X,Y) :- has(X,Y) -> has(X,Y)

This system deals with fact form hypotheses. The inverting resolution[Muggleton 88] method might be useful to generate rule form hypotheses. Our method generalizes the background theory and examples to make version spaces and specializes a least generalized support concept to explain a basic hypothesis. This reasoning is a kind of analogy.

7. Summary

This paper discussed a learning method about how hypothetical knowledge is extracted. This method uses an advanced reasoning which generates abductive explanations and selects hypotheses according to checking costs of hypothesis generation. A basic hypothesis is made by abduction when an explanation is not generated. A least generalized support concept is also derived from the basic hypothesis and version spaces which are made by a background theory and examples. This support concept is explained by induction of known knowledge. A assumptive macro knowledge is made from the explanation with this support concept. This knowledge is available unless the support knowledge is not denied. A current version of this method deals with facts as hypotheses. It is necessary for us to develop rule generation methods to explain examples and an integrated knowledge acquisition system (EPSILON/2) which consists of an interview system and a learning system.

8. Acknowledgment

We wish to express our thanks to the member of the Knowledge base System Architecture - Knowledge Acquisition and Design - Sub Working Group , to its head, Professor Riichirou Mizoguchi and to Professor Norihiro Abe. We would like to thank the members of the Learning and Non-monotonic Reasoning System Research Group at ICOT and Dr. Koichi Furukawa. We give thanks to the members of 5th laboratory at ICOT, Mr. Katsumi Inoue and Mr. Eiichi Horiuchi and the chief of 5th laboratory at ICOT, Mr. Kenji Ikoma. Finally, we give our thanks to Dr. Kazuhiro Fuchi, Director of ICOT Research

Center, who provided us with the opportunity to conduct this research in the Fifth Generation Computer Systems Project.

References

[Araragi 89] Araragi, T. and Ishida, T., Explanation-based Refinement of Knowledge Base, JSAI SIG-KBS-8805, 1989 (in Japanese)

[Atsumi 88] Atsumi, M., A View of EBL in a Hypothesis Reasoning System, Japan Software Science Society, Proc. 5th Conference, 1988 (in Japanese)

[Charniak 86] Charniak, E. and McDermott, D., Introduction to Artificial Intelligence, Addison Wesley, 1986

[de Kleer 86] de Kleer, J., An assumption-based TMS, Artificial Intelligence 28,pp.127-162, 1986

[Doyle 86] Doyle, R.J., Constructing and Refining Causal Explanations from an Inconsistent Domain Theory, AAAI-86, pp.538-544, 1986

[Falkenhainer 87] Falkenhainer, B., Forbus, K.D. and Gentner, D, The Structured-mapping Engine: Algorithm and Examples, Technical Report, UIUCDCS-R-87-1361, Dep. of Computer Science, University of Illinois, 1987

[Genesereth 86] Genesereth, M. and Nilsson, N., Logical Foundations of Artificial Intelligence, Los Altos, CA: Morgan-Kaufmann, 1986

[Inoue 89] Inoue, K., Problem Solving with Hypothetical Reasoning, 7th ICOT symposium proceeding, June, 1989 (in Japanese)

[Laird 86] Laird, J., Rosenbloom, P. and Newell, A., Universal Subgoaling and Chunking, Kluwer Academic Publishers, 1986

[Mitchell 77] Mitchell, T., Version spaces: A candidate elimination approach to rule learning, Proc. 5th IJICAI,pp.305-310, 1977

[Mitchell 85] Mitchell, T., Mahadevan, S. and Steinberg, L., LEAP: A Learning Apprentice for VLSI Design, Proc. 9th IJICAI, Los Angeles, 1985

[Mitchell 86] Mitchell, T., Keller, R., and Keder-Cabelli, S., Explanation-Based Generalization: A Unifying View, Machine Learning 1, 1986

[Muggleton 88] Muggleton, S. and Buntine, W., Machine Invention of First Order Predicates by Inverting Resolution, Proc. 5th Machine Learning, pp339-352, 1988

[Poole 88] Poole, D., A Logical Framework for Default Reasoning, Artificial Intelligence 36, pp.27-47, 1988

[Rajamoney 88] Rajamoney, S.A., Experimentation-Based Theory Revision, AAAI Spring Symposium Series: Explanation-Based Learning, pp. 7-16, 1988

[Shapiro 81] Shapiro, E.Y., Inductive inference of theories from facts, Technical Report 192, Yale University Computer Science Dept, 1981

[Stickel 88] Stickel, M.E., A Prolog-like Inference System for Computing Minimum-Cost Abductive Explanations in Natural-Language Interpretation, SRI International Tech.

Note 451, Sept. 1988

[Suwa 89] Suwa, M. and Motoda, H., Understanding Metaphor by Frustration-Based Learning, Proc. SICE 9th Knowledge Engineering Symposium, 1989 (in Japanese)

[Taki 87] Taki, H., Tsubaki, K. and Iwashita, Y., Expert Model for Knowledge Acquisition, Proc. IEEE 3rd Annual Expert Systems in Government Conference, Washington D.C., 1987

[Uchihashi 89] Uchihashi, T. and Taki, H., An Interactive EBG System, ICOT Technical Report, 1989 (to appear)

[Yamamura 87] Yamamura, M. and Kobayashi, S., An Integration of SBL and EBL based on Operationality Criteria, SICE Proc. 1st Knowledge Engineering Meeting, 1987 (in Japanese)

Call for Participation:
4TH KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP

Sponsored by the:
AMERICAN ASSOCIATION FOR ARTIFICIAL INTELLIGENCE (AAAI)

Banff, Canada
October 1 - 6, 1989

A problem in the process of building knowledge-based systems is acquiring and modeling appropriate problem-solving knowledge. The objective of this workshop is to assemble theoreticians and practitioners of AI who recognize the need for developing methods and systems that assist the knowledge acquisition process.

To encourage vigorous interaction and exchange of ideas the workshop will be kept small - about 40 participants. There will be individual presentations and ample time for technical discussions. An attempt will be made to define the state-of-the-art and future research needs. Attendance will be limited to those presenting their work, one author per paper.

Papers are invited for consideration in all aspects of knowledge acquisition for knowledge-based systems, including (but not restricted to):

o Transfer/modeling of expertise - systems that obtain and model knowledge from experts.
o Transfer/modeling of expertise - manual knowledge acquisition methods and techniques.
o Apprenticeship, explanation-based, and other learning systems; integration of such systems with other knowledge acquisitIon techniques.
o Issues in cognition and expertise that affect the knowledge acquisition process.
o Extracting and modeling of knowledge from text.
o Eliciting and modeling knowledge from multiple sources.
o Integration of knowledge acquisition techniques within a single system; integration of knowledge acquisition systems with other systems (hypermedia, database management systems, simulators, spreadsheets...).
o Knowledge acquisition methodology and training.
o Validation of knowledge acquisition techniques; the role of knowledge acquisition techniques in validating knowledge-based systems.

Five copies of a draft paper (up to 20 pages) should be sent to John Boose before May 1, 1989. Acceptance notices will be mailed by July 3. Full papers (20 pages) should be returned to the chairman by September 1, 1989 so that they may be bound together for distribution at the workshop.

There will be a travel-and-expense award of up to $500.00 US for the best paper submitted by a graduate student. Please note if the paper should be considered for this award.

Workshop Co-chairmen:

Send papers via US mail to:
John Boose
Advanced Technology Center
Boeing Computer Services, 7L-64
PO Box 24346
Seattle, Washington, USA 98124
(206) 865-3253
john@boeing.com

Brian Gaines
Department of Computer Science
University of Calgary
2500 University Dr. NW
Calgary, Alberta, Canada T2N 1N4
(403) 220-5901
gaines@calgary.cdn

Send papers via express mail to:
John Boose
Advanced Technology Center, 7L-64
Boeing Computer Services, Bldg. 33.07
2760 160th Ave. SE
Bellevue, Washington, USA 98008

Program Committee:

Jeffrey Bradshaw, Boeing Computer Services
B. Chandrasekaran, Ohio State University
William Clancey, Institute for Research on Learning
Michael J. Freiling, Tektronix, Inc.
Thomas Gruber, Stanford University
Catherine Kitto, Boeing Computer Services
Sandra Marcus, Boeing Computer Services
John McDermott, Digital Equipment Corporation
Ryszard Michalski, George Mason University
Mark Musen, Stanford University
Bruce Porter, University of Texas at Austin
Mildred Shaw, University of Calgary

FGCS 88で発表者.