CONF-9110444--2

Automated Knowledge Acquisition for Second Generation Knowledge Base Systems: A Conceptual Analysis and Taxonomy

by
Williams, K.E.
&
Kotnour, T.

Published in:
Proceedings for the 27th Annual Meeting of the Institute of Management Sciences, SE Chapter, Oct. 1991, Myrtle Beach, SC.

FG02-91NP00119

AUTOMATED KNOWLEDGE ACQUISITION FOR SECOND GENERATION KNOWLEDGE BASE SYSTEMS: A CONCEPTUAL ANALYSIS AND TAXONOMY

Tim Kotnour (Student), Virginia Tech
Kent Williams, Virginia Tech

ABSTRACT

In this paper, we present a conceptual analysis of knowledge-base development methodologies. The purpose of this research is to help overcome the high cost and lack of efficiency in developing knowledge base representations for artificial intelligence applications. To accomplish this purpose, we analyzed the available methodologies and developed a knowledge-base development methodology taxonomy. We review manual, machine-aided, and machine-learning methodologies. A set of developed characteristics allows description and comparison among the methodologies. We present the results of this conceptual analysis of methodologies and recommendations for development of more efficient and effective tools.

INTRODUCTION

The purpose of this paper is to help managers choose a knowledge acquisition method to aid problem solving by providing a taxonomy of knowledge acquisition methods.

This paper describes problem solving, the knowledge acquisition process, and knowledge acquisition methods. Problem solving is investigated because we must first understand problem solving before we can help managers solve problems. Knowledge acquisition is reviewed because it is the process used to elicit and organize the problem-space knowledge. Specific knowledge acquisition methods are investigated because they provide the means to perform knowledge acquisition. The knowledge acquisition methods are evaluated by their specific process, by their ability to directly map the problem-space representation, and on the level of a knowledge engineer's participation.

Managers and experts solve problems and make decisions. Managers and experts need knowledge to solve problems and make decisions. This knowledge is contained in a problem representation and problem space. The problem representation contains the initial states and goals of the problem. The problem space includes the goal and subgoals, possible intermediate states, operators which move the problem solver from state to state, and constraints on the problem [23]. This knowledge needs to be extracted, integrated, stored, and retrieved to aid problem solving.

Developing the problem representation and space is the most important part of the problem-solving process. The problem space is the foundation of problem solving because: 1) it represents the generation of all possible alternatives from which decisions must be made and 2) it shows the impact of a decision on other decisions that must be made. The problem space "consists of the information known or potentially available to the solver that may be useful in solving the problem" [23, p. 167]. It represents all the knowledge brought to bear in solving the problem. In fact, the problem space is a knowledge base, the "core rules and data that make up the domain knowledge" [8, p. 113]. To increase the efficiency and effectiveness of problem solving, tools must be built to elicit or extract the problem-space knowledge from experts and managers.

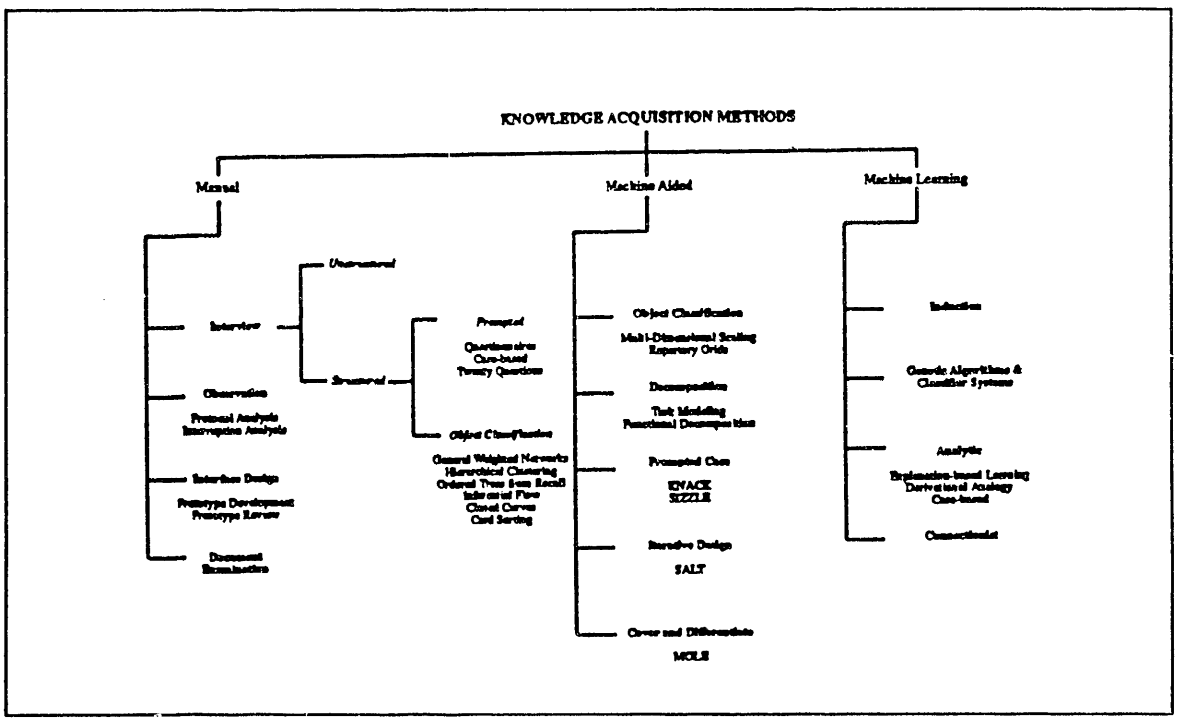
Knowledge acquisition is "the process that extracts knowledge from a source (e.g., a domain expert or textbook) and incorporates it into a knowledge-based system that solves some problem" [5, p. 65]. Knowledge acquisition is the iterative process by which knowledge is: 1) elicited, 2) organized, 3) represented, 4) refined, and 5) verified for use in a knowledge-based system to solve problems.

Knowledge elicitation involves acquiring or drawing the knowledge from sources (e.g., experts, case examples, or reference books). Knowledge organization takes the elicited knowledge and organizes it into an understandable and meaningful manner. Knowledge representation structures and formats the organized knowledge into a form, such as a production rule, acceptable by a specific knowledge-based system. Knowledge base refinement involves checking for inconsistencies, gaps in logic, conflicts, contradictions, and incompleteness in the knowledge base. Verification is done to verify the knowledge base is valid as compared to the real world.

KNOWLEDGE ACQUISITION METHODS

Based upon our analysis, knowledge acquisition methods are grouped into three categories: manual, machine-aided, and machine learning. From our analysis of the knowledge acquisition methods, a taxonomy was created. See Figure 1. The first level of the hierarchy (manual, machine-aided, and machine learning) classifies the methods based upon the degree of a knowledge engineer's interaction. The remaining levels of the hierarchy classify the methods by the specific means employed to perform the knowledge acquisition process.

FIGURE 1: TAXONOMY OF KNOWLEDGE ACQUISITION METHODS



Knowledge acquisition methods were investigated because they provide the means to capture an expert's knowledge (i.e., problem space). A single complete source of information on knowledge acquisition methods does not exist. Books, journal articles, proposals, and proceedings on knowledge acquisition were collected. The methods were analyzed on their methodology, correlation to the information processing model of problem solving, and required degree of a knowledge engineer's interaction. Methodology and knowledge engineer's participation are criteria because the means to accomplish the knowledge engineering process is a distinguishing characteristic of each method and can be used to group the methods. Correlation to the information processing model of problem solving is a criterion because we are searching for the best method to map the problem-space representation.

Manual Knowledge Acquisition

Manual methods require the knowledge engineer to be directly involved in the complete process. The knowledge must first be elicited and then manually organized. Once the knowledge is represented, the knowledge engineer must manually encode the knowledge in a form acceptable to a specific knowledge-base system. Finally, the refinement and verification is done by a manual step-by-step examination of the knowledge base and the knowledge-based system. There is little machine-aid used by the knowledge engineer while performing manual knowledge acquisition. Manual knowledge acquisition is divided into interview, observation, interface design, and document examination.

Interviews are conducted between the knowledge engineer and an expert or group of experts. The basic process is a question and answer session(s) between the knowledge engineer and expert. The sessions can be tape recorded and transcribed for analysis. Interviews are good starting points for knowledge acquisition. In the process, experts reveal relations between objects and the thought process used in solving a problem and designing a solution [18]. This process can be very time consuming [18]. Interviewing is also limited by the expert's ability to express himself in a meaningful way to the knowledge engineer. Interviewing is further broken down into structured and unstructured methods. This distinction is based upon the presence or lack of explicit structure placed on the interviewing process.

Structured interviewing methods use closed questions in a structured plan [24]. The knowledge engineer needs to have domain knowledge to provide structure better. Structured interview methods include prompted interview and object classification.

Prompted interview methods, the first set of structured interview methods, use a prompt to guide the expert's response. The structure is in the process chosen to prompt the expert. Case-based, questionnaires, and twenty questions are examples of this method [18] [24]. When using a case-based interview the expert is asked to solve a domain case or example problem developed by another expert. Questionnaires allow the expert to answer the knowledge engineer's question at his or her leisure. To do twenty questions, the knowledge engineer chooses a domain object and the expert asks *twenty* yes-no questions trying to determine the object.

Object classification, the second set of structured interview methods, is used to classify or group objects or concepts. Object classification methods include: general weighted networks, hierarchical clustering, ordered trees from recall, inferential flow, closed curves, and card sorting [18] [24]. Each of these methods provides the structure to the interviewing process by asking specific questions about objects or items in the area of investigation. The expert is asked to describe relations among objects and this information is analyzed to produce a classification of the objects. These methods are used to elicit the expert's clustering, relationships, structure or organization of domain objects.

The unstructured interview, as its name implies, uses very general and open-ended questions [24]. However, this eventually leads into a structured interview as the knowledge engineer learns more about the domain. The unstructured interview is useful for initial knowledge acquisition.
Observation is the second set of manual methods. The expert performs the task while the knowledge engineer watches and/or videotapes the process [9]. A transcript is created from a videotape. These methods are useful for identifying problem-solving strategies, studying motor skills, and verifying experts' task descriptions [24]. Observation methods include protocol analysis and interruption analysis [18]. Protocol analysis is used to catch the expert in the act of performing the task being analyzed [9]. Protocol analysis is done as the expert performs the task. He or she describes aloud what he or she is doing. Interruption analysis follows the same basic process as protocol analysis. However, the expert performs the task until the knowledge engineer doesn't understand what the task is, at this time the expert describes the task aloud [18].

Interface design is the third set of manual knowledge acquisition methods. Interface design includes prototype development and review [24]. These methods have the knowledge engineer and expert working together in describing and evaluating a prototype knowledge-based system.

Document examination is the last set of manual knowledge acquisition methods. This is very useful for both initial and detailed knowledge acquisition of domain theory and principles. Reference books such as troubleshooting manuals provide a wealth of information to start system development. Some document examination should be done before working with experts so the knowledge engineer can gain a good initial understanding of the domain.

Machine-Aided Knowledge Acquisition

Machine-aided methods interactively elicit, organize, compile, and refine the knowledge with the knowledge engineer. The knowledge engineer and machine interactively elicit the domain knowledge. As the elicitation process proceeds, the machine organizes the knowledge. Once the knowledge is elicited and organized, the machine automatically represents the knowledge. Most machine-aided tools provide facilities to interactively refine the knowledge base. Machine-aided methods guide the knowledge acquisition process on its own. Machine-aided methods are typically automated versions of the manual methods. The automation aids in the organization, representation, and refinement phases. By using these tools, experts can directly perform the knowledge acquisition process with little or no help from a knowledge engineer. Machine-aided tools are divided into object classification, decomposition, prompted/case, interative design, and cover-and-differentiate.

As in the manual object classification, machine-aided object classification methods involve interviewing the expert about relationships among objects. Two machine-aided object classification tools are multi-dimensional scaling and repertory grids [4] [18] [24]. Multi-dimensional scaling groups objects based on their relative distances from each other. Repertory grids, based on personal construct theory, generates rules for classification-diagnosis problems.

Machine-aided decomposition attempts to decompose an object or concept into its components. Machine-aided decomposition tools include task modeling and functional decomposition [7] [20]. Task modeling is used by Di Piazza [7] in a tool to interview the expert about assumptions and goals of the expert system. Functional decomposition is used by Pugh and Price [20] in their tool to interview an expert about the composition of an object, its components, and their interrelated states.

Machine-aided prompted case methods use an expert case description, interactively adapting it to solve a problem. Tools using this technique include KNACK [13] and SIZZLE [17]. KNACK uses three items: a sample report, a domain model, and strategies for acquiring knowledge to aid report writing. SIZZLE uses past cases and extrapolation knowledge to solve a sizing problem.

Iterative design methods aid the expert in the design process by organizing and compiling the gathered design knowledge. SALT is an example of an iterative design tool. SALT uses a propose-and-revise problem-solving strategy to aid in the design constraint satisfaction processes [14].

Cover and Differentiate is a problem-solving process in which the expert specifies 1) candidates to cover a problem and 2) information used to differentiate the candidates [10]. MOLE is an automated knowledge acquisition tool that uses the cover-and-differentiate strategy to diagnose problems [10].

Machine-Learning Knowledge Acquisition

The machine-learning methods require very little direct interaction of the knowledge engineer. He or she is responsible for providing or gathering the data to be used by the specific machine-learning process. These methods automatically organize, represent, refine, and verify the knowledge base. The machine determines the content of the knowledge-base representation. The knowledge engineering process is fully embedded in the software. Machine learning has four paradigms: induction, genetic algorithms, analytic, and connectionist [6].

Induction learns rules by "inducing a general concept description from a sequence of instances of the concept and (usually) known counter-examples of the concept" [6, p. 3]. The purpose is to build the description so the positive examples can be later classified and the negative examples not classified [11]. Induction is used primarily for classification due to the decision tree created.

Genetic algorithms based on the genetics of biology are used to increase the performance of classifier systems by discovering rules. Classifier systems are most useful when there is a continuous stream of environmental data, need for real time action, inexactly defined goals, and little reinforcement [3] [19].

Analytic machine learning methods improve the efficiency of a system by using past problem solving examples and/or domain theory [6]. Analytic methods include explanation-based learning, derivational analogy, and case-based methods. Explanation-based learning uses domain theory and a concept example to acquire search control rules from a problem-solving trace to improve problem solving [1] [15]. Derivational analogy uses top-down decomposition to conduct a new plan based upon previous designs. A design plan is replayed to solve the new case by selecting and adapting it to fit the new case [2] [16]. Case-based methods first explain why previous knowledge or expectations failed to apply and then they correct expectations [22].

The connectionist paradigm uses neural networks to develop rules for pattern recognition. Neural networks have input, hidden, and output layers. The input and output layers are coded to define what each input and output neuron represent. The hidden layer is where the work of the network takes place. The network contains units, neurons, with interconnection weights among them and a transformation function to transform a unit's inputs to its output [12] [21].

CONCLUSIONS

A taxonomy of knowledge acquisition methods was created from our analysis. Not one of the methods reviewed directly and explicitly mapped the problem-space of problem solving. These methods are either too generic or too specific. Also, we believe a machine-aided tool can provide the most benefit to knowledge acquisition for problem solving. By using a machine-aided tool, an expert or group of experts doesn't need to rely on a knowledge engineer and can directly construct an integrated problem-space representation.

Our goal is to develop a tool to directly and explicitly perform knowledge acquisition of a problem-space representation. The next step is to research cognitive task analysis methods to determine if there is a method to help us meet our purpose.

ACKNOWLEDGEMENTS

The preparation of this paper is funded by the Office of Naval Research Cognitive Sciences Program, Contract No. N00014-91-J-1500.

REFERENCES

[1] Adeli, H. and Yeh, C., "Explanation-based machine learning in engineering design." Engineering Applications of Artificial Intelligence, vol. 3, No. 2 (June, 1990), pp.127-37.

[2] Bhansali, S. and Harandi, M.T., "Program synthesis using derivational analogy" (Tech. Rep. No. UIUCDCS-R-90-1591 Urbana-Champaign: Univ. Illinois at Urbana-Champaign, Department of Computer Science, 1990).

[3] Booker, L.B., Goldberg, D.E., and Holland, J.H., "Classifier systems and genetic algorithms." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp.235-282) (Cambridge, Massachusetts: The MIT Press, 1990).

[4] Bradshaw, J.M. and Boose, J.H. (1990). "Decision analysis techniques for knowledge acquisition: Combining information and preferences using Aquinas and Axotl." International Journal of Man-Machine Studies, vol. 32, No. 2 (February, 1990), pp. 121-86.

[5] Bylander, T. and Chandrasekaran, B., "Generic tasks in knowledge-based reasoning: The 'right' level of abstarction for knowlegde acquisition." In B.R. Gaines and J.H. Boose (Eds.), Knowledge acquisition for knowledge-based systems (pp. 65-77) (San Diego, California: Academic Press Inc., 1988).

[6] Carbonell, J.G., "Introduction: Paradigms for machine learning." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp. 1-9) (Cambridge, Massachusetts: The MIT Press, 1990).

[7] Di Piazza, J.S., "Interweaving knowledge extracting, organizing and evaluating: A concrete design for preventing logic and structure bugs while interviewing experts." Journal of Automated Reasoning, vol. 6, No. 3 (1990), pp. 299-317.

[8] Dym, C.L. and Levitt, R.E., Knowledge-based systems in engineering (McGraw Hill, Inc., 1990).
[9] Ericson, K.A. and Simon, H.A., Protocol analysis: Verbal reports as data (Cambridge, Massachusetts: The MIT Press, 1984).
[10] Eshelman, L., "MOLE: A knowledge-acquistion tool for cover-and-differentiate systems." In S. Marcus (Ed.), Automating knowledge acquisition for experts systems (pp. 37-80) (Boston, Massachusetts: Kluwer Academic Publishers, 1988).
[11] Gennari, J.H., Langley, P., and Fisher, D., "Model of incremental concept formation." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp. 11-61) (Cambridge, Massachusetts: The MIT Press, 1990).
[12] Hinton, G.E., "Connectionist learning procedures." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp. 185-234) (Cambridge, Massachusetts: The MIT Press, 1990).
[13] Klinker, G., "KNACK: Sample-driven knowledge acquisition for reporting systems." In S. Marcus (Ed.), Automating knowledge acquisition for experts systems, (pp. 125-174) (Boston, Massachusetts: Kluwer Academic Publishers, 1988).
[14] Marcus, S., "SALT: A knowledge-acquisition tool for propose-and-revise systems." In S. Marcus (Ed.), Automating knowledge acquisition for experts systems (pp. 81-123) (Boston, Massachusetts: Kluwer Academic Publishers, 1988).
[15] Minton, S., Carbonell, J.G., Knowblock, C.A., Kuokka, D.R., Etzioni, O., and Gil, Y., "Explanation-based learning: A problem solving perspective." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp.63-118) (Cambridge, Massachusetts: The MIT Press, 1990).
[16] Mostow, J., "Design by derivational analogy: Issues in the automated replay of design plans." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp. 119-184) (Cambridge, Massachusetts: The MIT Press, 1990).
[17] Offutt. D., "SIZZLE: A knowledge-acquisition tool specialized for the sizing task." In S. Marcus (Ed.), Automating knowledge acquisition for experts systems (pp. 175-200) (Boston, Massachusetts: Kluwer Academic Publishers, 1988).
[18] Olson Reitman, J. and Rueter, H.H., "Extracting expertise from experts: Methods for knowledge acquisition." Expert Systems, vol. 4, No. 3 (August, 1987), pp.152-168.
[19] Pettit, E and Pettit, M., "Analysis of the performance of a gentic algorithm-based system for message classification in noisy environments." In B.R. Gaines and J.H. Boose (Eds.), Knowledge acquisition for knowledge-based systems (pp. 335-350) (San Diego, California: Academic Press Inc., 1988).
[20] Pugh, D.R.and Price, C.J., "Automating knowledge acquisition for generic tasks." Engineering Applications of Artificial Intelligence, vol. 3, No. 3 (September, 1990), pp. 171-9.
[21] Rumelhart, D.E., Hinton, G.E., and McClelland, J.L., "A general framework for parallel distributed processing." In J.L. McClelland, D.E. Rumelhart, and the PDP Research Group (Eds.), Parallel distributed processing (pp. 45-76) (Cambridge, Massachusetts: The MIT Press, 1989).
[22] Schank, R.C. and Leake, D.B., "Creativity and learning in a cased-based explainer." In J. Carbonell (Ed.), Machine learning: Paradigms and methods (pp. 353-385) (Cambridge, Massachusetts: The MIT Press, 1990).
[23] Voss, J.F., Greene, T.R., Post, T.A., and Penner, B.C., "Problem solving skill on the social sciences." In G.H. Bower (ed.), The psychology of learning and motivation: Advances in research theory (Vol. 17) (pp. 165-212) (New York: Academic Press, 1983).
[24] Welbank, M., "An overview of knowledge acquisition methods." Interacting with Computers, vol. 2, No. 1 (1990), pp. 83-91.

END

DATE
FILMED
7/8/93