

AUTOMATED KNOWLEDGE ACQUISITION FOR BUILDING DIAGNOSIS EXPERT SYSTEMS

Thomas M. Gatton, Computer Engineer
and
Frank W. Kearney, Team Leader

U.S. Army Corps of Engineers
Construction Engineering Research Laboratory
P.O. Box 4005
Champaign, Illinois 61820

ABSTRACT

The knowledge acquisition process in traditional knowledge engineering methods has been identified as the bottleneck in the development of expert systems. A system called the Interactive Domain Model (IDM) has been developed to automate the development of diagnostic expert systems. The IDM embodies meta-knowledge about the diagnosis of physical systems, such as buildings and machines, and views a particular model through a Graphical Abstract Domain Model (GADM). A GADM is a hierarchical model of the system composed of subassemblies and components. This model is utilized in a structured query of the expert for information about the system. This information is then used to automatically generate a knowledge base for a particular shell, thereby eliminating both the knowledge acquisition bottleneck and the manual interpretation of that knowledge into a knowledge base. This substantially reduces the amount of time necessary for the development of an expert system. Presently, research is being conducted to extend this concept to the realm of systems inspection and maintenance, to include both civil works and facilities, as well as quality assurance in construction. Long term goals include further extension into other areas of design, construction, and systems operation.

1. BACKGROUND

Over the past eight years, the Materials and Quality Assurance team of the Engineering and Materials Division at the Construction Engineering Research Laboratory has developed the following expert systems:

- 1) WELDER - Non-destructive evaluation of welds,
- 2) NDE - Condition assessment of concrete,
- 3) ESRAM - Railway diagnosis and maintenance,
- 4) DRH AC - Diagnosis and repair of residential heating, ventilating, and air conditioning equipment,
- 5) ESROM - Diagnosis, maintenance, and repair of Built-Up-Roofs,
- 6) MITER - Diagnosis and repair of miter gates,
- 7) ESAP - Inspection and diagnosis procedures for asphalt paving, and
- 8) ESROC - Quality assurance in built-up-roof construction.

Most of these expert systems were developed using the CRITIC shell that was developed in-house by the MQA team before commercial shells were available. Evaluation of both user requirements and available shells is an ongoing process to insure that the right tool is being used for a particular application. User requirements usually dominate and, as a result, the CRITIC shell is most often used due to the fact that the source code can be altered to customize the user interface and add new features to the shell as needed.

The traditional development of an expert system must involve one of two different approaches. Either a knowledge engineer must become conversant with the domain or the expert must be educated about the development of expert systems to the point that he knows enough about knowledge engineering to structure his knowledge in the syntax of the shell system. Generally, though, experts are not computer oriented and, further, cannot be expected to take time to become knowledge engineers. The current MQA team approach in the development of an expert system is for the knowledge engineer to develop a prototype system prior to any contact with the expert(s). The prototype contains "book knowledge" that is commonly available about a particular application. This allows the knowledge engineer to become conversant with a particular domain thereby eliminating the need for the expert to "tutor" the knowledge engineer from ground zero, or thereabouts. This is advantageous because it brings the knowledge engineer to a level of domain knowledge that allows communication with the expert about the subject matter. Also, it provides an initial system which the expert can immediately begin to correct and verify, thereby reducing the time requirements for application development.

Although this approach improves the efficiency of the knowledge engineering process, the bottleneck of knowledge acquisition and the necessity for the knowledge engineer to become a "pseudo expert" remain as major barriers to rapid development. Solutions to this problem must be developed and the automation of knowledge acquisition would provide a major reduction in development time.

2. OBJECTIVES

The major objective of this research is to develop a method to automate the knowledge acquisition for diagnostic expert systems and allow direct input of domain knowledge by an expert. This expertise would then be used to automatically generate a knowledge base. Such a system would eliminate the need for traditional knowledge engineering and introduce the new role of the knowledge engineer in this technology, the Meta-Knowledge Engineer. This is evident by the fact that the algorithms utilized to gather information directly from experts embody knowledge about the diagnosis of systems. An additional objective of this research is to investigate the concepts of meta-knowledge engineering and their applicability beyond diagnostics and into thought and decision making processes embodied in design, construction, operations, and maintenance of systems.

3. METHODOLOGY

The use of abstract domain models has been investigated as a potential user interface. Recent research [LANGE86, GATTON87, BUCHNER88] has led to two possible application of these models in automated knowledge acquisition. This research deals with the use of a graphical abstract domain (GADM) as the user/expert interface and the algorithms to link that model to the production rule representation of the domain knowledge and its associated heuristics.

A graphical abstract domain (GADM) is a simple hierarchical representation of a physical system broken down into its subassemblies and components. Presently, a GADM is limited to element representation of three types:

- 1) assembly,
- 2) subassembly, and
- 3) component.

This scheme is demonstrated in Figure 1 where a partial breakdown an automotive engine is shown. It is labelled to indicate assembly (A), sub-assembly (S), and component (C) elements of the model and

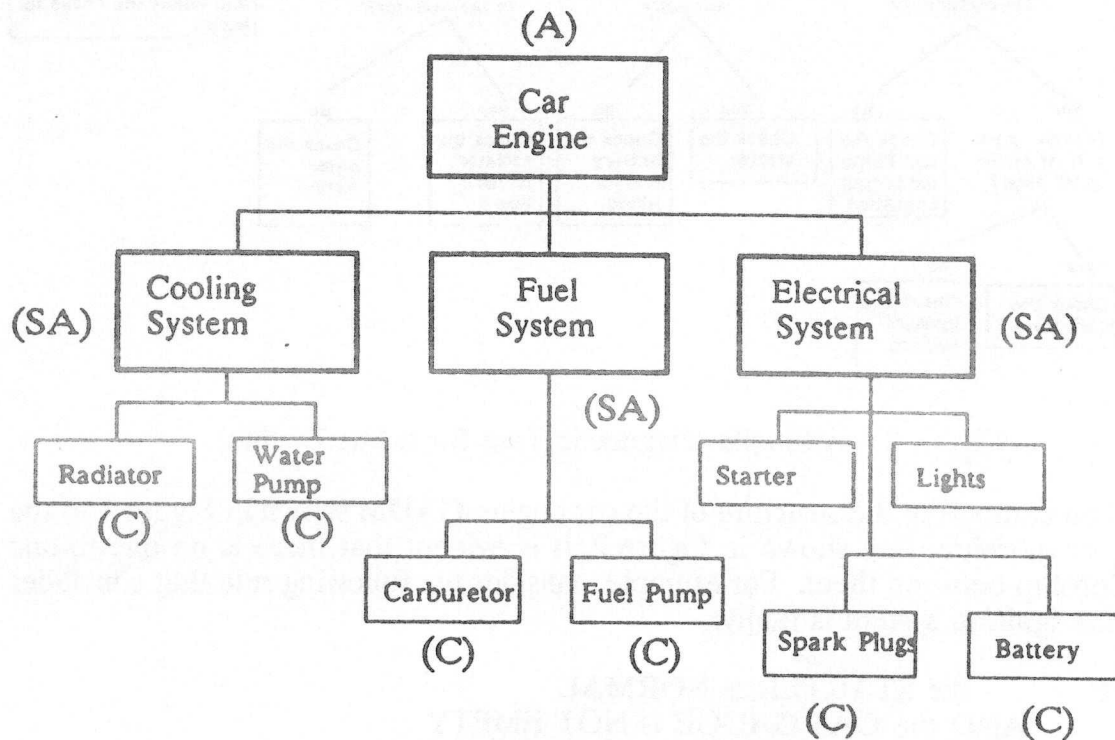


Figure 1 - A Partial GADM for a Car Engine

structured in a hierarchy to represent a simplified GADM of the levels and organization of elements that are present in an automobile engine. By definition, the element at the top level of the GADM is always the assembly, the elements at the bottom of the GADM are the components, and any elements in between are subassemblies.

The GADM shown in Figure 1 is related to the domain knowledge and heuristics that an expert mechanic has which is minimally represented by the logic, or decision, tree shown in Figure 2. This tree embodies the production rules that would be utilized to diagnose an engine that does not start or overheats.

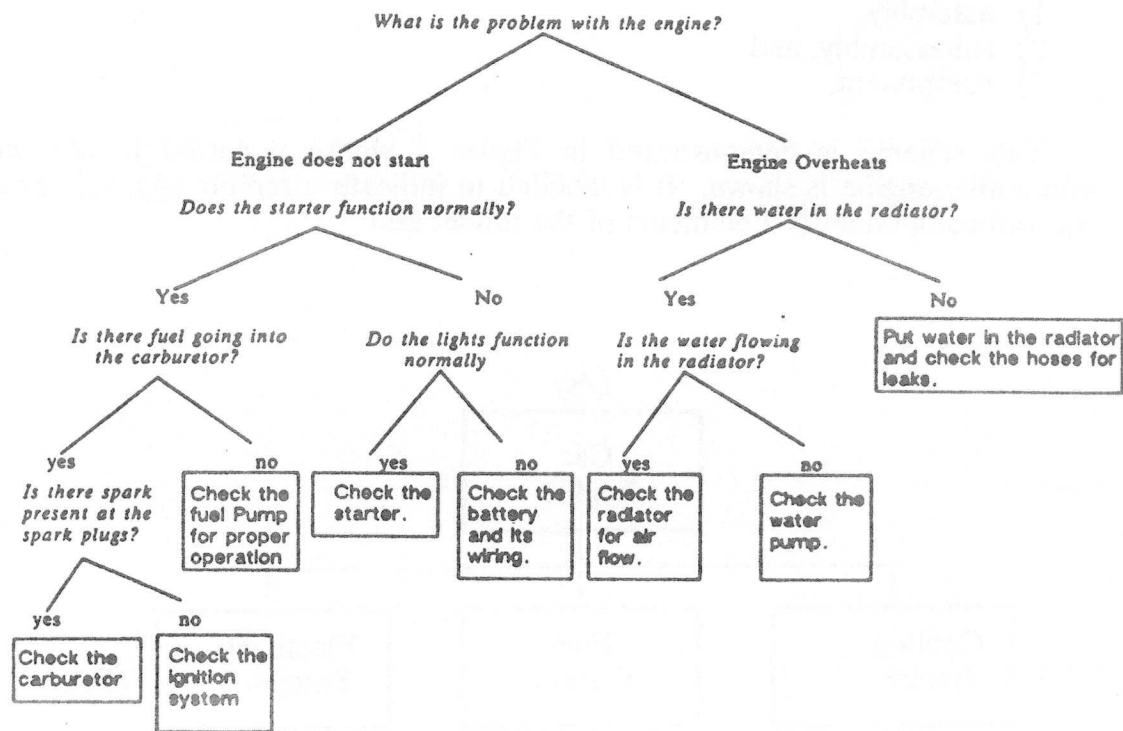


Figure 2 - A Simple Diagnostic Tree for a Car Engine

When comparing the structure of the car engine GADM shown in Figure 1 to the logic, or decision, tree shown in Figure 2, it is evident that there is no one-to-one relationship between them. For example, consider the following rule that concludes that the ignition system is faulty:

IF the STARTER is NORMAL
 AND the GAS GAUGE is NOT EMPTY
 AND the CARBURETOR is WORKING
 AND the SPARK PLUG is SPARKING
 THEN CHECK THE IGNITION SYSTEM

Upon tracing this rule through the car engine GADM, it can be seen that the rule jumps between the fuel system and the electrical system in order to determine the faulty component. This indicates that the expert is trying to identify faulty components or eliminate other components as possible causes for a problem condition. In order to capture the domain knowledge and heuristics that the expert has, the relationship between the GADM and the logic tree must be identified. This involves identifying the possible conditions of the components, the symptoms of the assembly and subassemblies, and the dynamic logic involved under the different combinations of symptoms and conditions.

Once the expert has built a GADM to represent the system, information concerning the types of questions asked and the possible answers that can be given to determine the condition of each component are obtained from the expert. Along with each of the possible conditions, any possible conclusion and recommendation for repair are gathered. Finally, for this particular component failure, the symptom present in the assembly itself is identified by the expert. This is the real key in organizing and is due to the fact that a faulty component exhibits particular symptoms in the assembly. It is notable, however, that this is only true in diagnosis and not in inspection and maintenance, where components have not yet reached a condition that exhibits an assembly symptom. After all of the components' information has been gathered, all of the assembly symptoms are known, assuming a complete GADM. With this information, queries can begin at the assembly by asking for a given assembly symptom what should be checked next. The next check will be either a subassembly or a component. If it is a subassembly, then information about that subassembly's conditions will be gathered. This will be repeated in a combinatoric fashion until all of the rules and their associated conclusions have been identified. This information contains the domain knowledge and heuristics gathered from the expert. This knowledge is then used to automatically generate a knowledge base. Specific translators can be used to produce the correct syntax for a given shell system.

The expert can modify the model and the logic through other options in the program. The expert is able to add, modify, and delete information about the questions, the conclusions and recommendations, and the rules used to reach the various conclusions.

4. PROJECT STATUS

Currently, a prototype system has been developed that will allow the construction of a GADM and interact with the expert to gather the domain knowledge and heuristics about diagnosing faulty components in that model. This information can be modified and added to at a later time. The options that allow changes in the GADM and then interactively query the expert for information to update the logic tree are not yet complete. Initial domain testing has indicated that a wide range of diagnostic applications can be handled by the program.

5. PRELIMINARY RESULTS

The prototype has indicated the feasibility of developing algorithms to link a graphical model (GADM) with the respective logic tree that embodies diagnostic procedures. The program has high potential as a tool to automate the knowledge acquisition process as well as eliminate the need for domain training of the knowledge engineer and conversion of expertise into a knowledge base. Application of the IDM to domains including inspection and maintenance is being conducted in order to identify new or modified algorithms for gathering the respective

An interesting fact about the GADM is that it can be understood by all disciplines across the whole spectrum of design, construction, operations, and maintenance. This leads to the notion that it may be utilized as a fundamental modelling scheme allowing data integration through the life cycle of the system. It is also evident that generic graphical models can be built to cover a range of possible designs and a particular design will trigger the appropriate parts of the logic trees. In quality assurance, for example, a particular built-up-roof design may vary in the deck, vapor barrier, insulation, and bitumen type but for each of these specifications there exists a unique set of checks that are performed to insure that construction is being performed properly. Initial investigations indicate the flexibility of the graphical model and its potential to be utilized across the entire spectrum of a product's life cycle.

Work is continuing in this area as well as extending the types of knowledge representation that are available for knowledge acquisition. Another area of interest is the use of a natural language interface for interaction. One of the limitations of the prototype is that many of the queries must be answered in a "forced phrase" mode. In other words, the expert must answer a question as if it was a phrase beginning with a given phrase. This is somewhat awkward for the expert and natural language is a possible solution. These areas offer strong potential for automated knowledge acquisition and offer challenging opportunities for future research.

REFERENCES

Buchner, B.A., "A Domain Theory for Knowledge Acquisition," Master's Thesis, University of Illinois, Department of Computer Science, May, 1988.

Gaines, Brian R., "Knowledge Acquisition: Developments and Advances," in Expert Systems and Intelligent Manufacturing, North-Holland, New York, 1988.

Gatton, T. M., "Decision Models for Expert System Development: Designing for Multi-Level User Domain Knowledge," Master of Computer Science Project, University of Illinois, 1987.

Harandi, M., Lange, R., & Kearney, F., "Distributed Knowledge Acquisition Using Abstract Domain Models," USA- CERL/ASCE Expert Systems Conference, June, 1988, Champaign, Illinois.

Lange, R., & Harandi, M.T., "The elements of a Distributed Knowledge Acquisition System," Proceedings of the 6th International Workshop on Expert Systems and Their Applications, April, 1986, Avignon, France.