AI Magazine Volume 8 Number 2 (1987) (© AAAI)

Knowledge Acquisition in the Development of a Large Expert System

David S. Prerau

This article discusses several effective techniques for expert system knowledge acquisition based on the techniques that were successfully used to develop the Central Office Maintenance Printout Analysis and Suggestion System (COMPASS) Knowledge acquisition is not a science, and expert system developers and experts must tailor their methodologies to fit their situation and the people involved. Developers of future expert systems should find a description of proven knowledge-acquisition techniques and an account of the experience of the COMPASS project in applying these techniques to be useful in developing their own knowledge-acquisition procedures

K*nowledge acquisition* is the process by which expert system developers find the knowledge that domain experts use to perform the task of interest. This knowledge is then implemented to form an expert system. The essential part of an expert system is its knowledge, and therefore, knowledge acquisition is probably the most important task in the development of an expert system.

In this article, several effective techniques for expert system knowledge acquisition are discussed based on the techniques that were successfully used at GTE Laboratories to develop the COMPASS expert system. Knowledge acquisition for expert system development is still a new field and not (yet?) a science. Therefore, expert system developers and the experts they work with must tailor their knowledge-acquisition methodologies to fit their own particular situation and the people involved. As expert system developers define their own knowledge-acquisition procedures, they should find a description of proven knowledge-acquisition techniques and an account of the experience of the COMPASS developers in applying these techniques to be useful.

The next section of this article is a discussion of the COMPASS project. The major portion of the article follows, with over 30 points on knowledge acquisition that were found to be important during the work on COMPASS. Initial points cover the knowledge-acquisition considerations in selecting an expert and an appropriate domain for the expert system. The remaining points highlight techniques for getting started in knowledge acquisition, documenting the knowledge, and finally, actually acquiring and recording the knowledge. Each point is followed by a general discussion and then by a description of how the point specifically applied to the COMPASS project.

COMPASS

COMPASS is a multiparadigm expert system developed by GTE Laboratories for telephone switching-system maintenance (Prerau et al. 1985b; Goyal et al. 1985). COMPASS accepts maintenance printouts from telephone company central office switching equipment and suggests maintenance actions to be performed.

In particular, COMPASS accepts maintenance printout information from a GTE Number 2 Electronic Automatic Exchange (No. 2 EAX). A No. 2 EAX is a large, complex telephone call switching system ("switch") that can interconnect up to 40,000 telephone lines. Such a switch generates hundreds or thousands of maintenance messages daily. The current manual procedure of analyzing these messages to determine appropriate maintenance actions takes a significant amount of time and requires a high level of expertise. COMPASS uses expert techniques to analyze these messages and produce a prioritized list of suggested maintenance actions for a switch-maintenance technician.

COMPASS is implemented on Xerox 1108 Lisp machines using the KEE™ system (Fikes and Kehler 1985) from IntelliCorp. The COMPASS implementation utilizes multiple artificial intelligence paradigms: rules, frame hierarchies, demon mechanisms, object-oriented programming facilities, and Lisp code.

COMPASS is a large expert system: the COMPASS "knowledge document" (Prerau et al., 1986), which contains a succinct English-language record of the COMPASS expert knowledge, is approximately 200 pages long. The COMPASS implementation consists of about 500 Lisp functions, 400 KEE rules, and 1000 frames with a total of 15,000 slots. The system (COMPASS, KEE, and Interlisp-D) requires about 10 megabytes. COMPASS alone requires about 5 megabytes, and is growing larger as data are analyzed.

In its initial field uses, COMPASS has displayed performance comparable to (and, in some cases, better than) that of domain experts and significantly better than that of average No. 2 EAX maintenance personnel (Prerau et al., 1985a). COMPASS is probably one of the first major expert systems designed to be transferred completely from its developers to a separate organization for production use and maintenance. COMPASS has been put into extensive field use by GTE Data Services (GTEDS) of Tampa, Florida (Prerau et al., 1985d). It has been run on a daily basis for about a year to aid maintenance personnel at 12 No. 2 EAXs in four states. These switches service about 250,000 telephone subscribers. COMPASS is currently being put into production use by GTE telephone companies.

Because COMPASS is designed to be maintained by a group completely separate from its developers, major consideration during development was given to the potential maintainability of the final COMPASS system. The COMPASS project team developed a set of software engineering techniques for expert system implementation (Prerau et al., 1987). These techniques were utilized for COMPASS and are being used in other expert system developments.

Selecting an Expert

A domain expert is the source of knowledge for the expert system. Therefore, even before the actual process of knowledge acquisition begins, a decision crucial to its success must be made: the choice of the project's expert (or experts). Because of the significance of this decision, among the important criteria for selecting an appropriate expert system domain are considerations related to the choice of a domain expert. These considerations primarily relate to the degree that the expert will function well in the role of knowledge source.

Importance of Expert Selection

- Significant time and effort is needed to select an expert.

The selection of an expert is an important element in knowledge acquisition, and knowledge acquisition is critical to the overall expert system.

Early in the COMPASS project, an extensive set of criteria for selecting an expert system domain were developed (Prerau 1985). This set included criteria for selecting a project expert (of these criteria, only those related to knowledge acquisition are discussed here). We then spoke with several contacts in the domain area and explained our need for a project expert and our criteria for selecting one. The discussions yielded a small list of potential No. 2 EAX experts for our project. The most promising of these experts were asked to come separately to GTE Laboratories for two days of meetings. At these meetings, we discussed the project, expert systems in general, and the potential participation of the expert in our project. At the same time, we tried to see how the potential expert met our selection criteria. Based on these meetings, we selected the COMPASS expert.

An Expert's Capabilities

- Select an expert who has developed domain expertise by task performance over a long period of time.

The expert must have enough experience to be able to develop the domain insights that result in heuristics (rules of thumb). These heuristics most distinguish the knowledge in an expert system from that in a conventional program and are the main goal of the knowledge-acquisition process.

Our COMPASS expert, W. (Rick) Johnson, is a switching-services supervisor in the electronic operations staff at General Telephone of the Southwest (GTSW). He has been working in telephone switching for 16 years, including about 5 years specifically on the No. 2 EAX.

- Select an expert who is capable of communicating personal knowledge, judgment, and experience and the methods used to apply these elements to the particular task.

An expert should not only have the expertise but also the ability to impart this expertise to the project team, whose members probably know little or nothing about the subject area. Experts should be introspective, able to analyze their reasoning processes; and communicative, able to describe those reasoning processes clearly to the project team.

The COMPASS expert was an excellent communicator in teaching the COMPASS knowledge engineering team the basics of the No. 2 EAX and in discussing and explaining the methods he used to analyze No. 2 EAX maintenance messages.

- An expert should be cooperative.

An expert should be eager to work on the project or at worst be nonantagonistic. It is a hard job to be a project expert and to have to examine in detail the way you have been making decisions. If the expert is not interested or is even resentful about being on the project, then the expert might not put in the full effort required. One way to ensure a cooperative expert is to find a person who is interested in computers and in learning about expert systems (and possibly in becoming a local "expert" on expert systems and AI when the project is completed). Also, an expert who sees a big potential payoff in the expert system being developed might want to be involved with it.

The COMPASS expert was very interested in, and enthusiastic about, the project, and the effort he put in was more than what was expected. He learned a good deal about AI and expert systems during his work on the COMPASS project and became familiar with the Lisp machines being used. Also, he received considerable visibility with his local management and eventually shared in a major award for COMPASS.

- Select an expert who is easy to work with.

A domain expert in an expert system project spends a lot of time with the project team.

In COMPASS, we had an excellent working relationship with our expert.

An Expert's Availability and Support

- Select an expert who is able to commit a substantial amount of time to the development of the system.

Because knowledge acquisition requires long hours, days, and weeks of discussions between experts and knowledge engineers, an expert for an expert system project must be willing and able to commit the significant time and effort required by the project.

One important factor in the COMPASS project was that our expert was willing and able to make a major commitment to the project.

- Strong managerial support is needed for an expert's commitment to the project.

Because knowledge acquisition for a major expert system can require many weeks or months of discussions with an expert, ample time should be set aside in the expert's schedule for meetings. Available time is often a problem. The best experts in the most important corporate areas are usually the ones who can least be spared from their usual position.

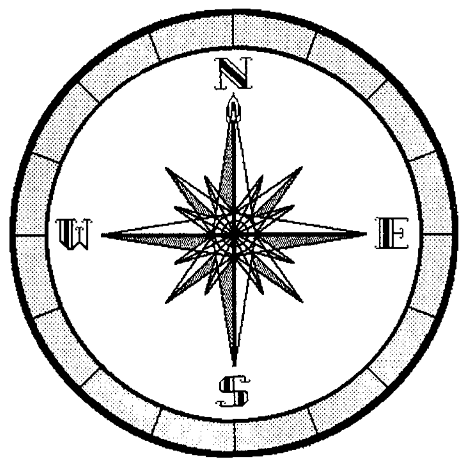
We were very fortunate in the COMPASS project to be able to obtain from GTSW management a commitment of one week per month of our expert's valuable time for the duration of the project (over two years). Any smaller commitment of time would have significantly affected the speed of the project development, and a major cutback would have made it almost impossible to achieve the results that we did.

Selecting the Domain

In addition to the selection of the expert, criteria for the selection of an appropriate domain for expert system development (Prerau 1985) directly relate to the ease of knowledge acquisition.

- The domain should be such that the expert system does not have to perform the entire task to be useful: some degree of incomplete coverage can be tolerated (at least initially).

If this statement is true, the expert system development project can begin by developing a system to cover one subdomain and then expand by adding other subdomains This method of development allows the knowledge acquisition for a large domain to be focused on one subdomain at a time.



For COMPASS, we spent almost the entire first year concentrating on one class of No. 2 EAX error messages (albeit the most important and most complex message class)--the "network recovery 20" messages. Our knowledge-acquisition sessions did not even consider the analysis of any other message types until our first system was completed. The subsequent expansion of COMPASS added the capability of handling every other No. 2 EAX message type that requires detailed expert analysis.

- The task should be decomposable, allowing relatively rapid prototyping for a closed small subset of the complete task and then slow expansion to the complete task.

This approach allows knowledge acquisition to focus on a subset of the task rather than the entire task at once. Combined with the previous item, the knowledge acquisition can then be directed at any one time to one subtask for one subdomain.

In COMPASS, we (including our expert) did not know at first that the task was decomposable; so, we started by finding rules and procedures for the entire initial task (analysis of network recovery 20 messages). After some time, it became clear that the task could be decomposed into five major phases: Input, Identify, Analyze, Suggest, and Output (Prerau et al, 1985b). Then, we were able to concentrate our knowledgc acquisition at any one time on one particular phase, thus focusing our attention.

Central Office Maintenance Printout Analysis and Suggestion System

- The domain should be fairly stable.

An unstable domain can yield a situation where a large number of knowledge structures (for example, rules) found early in the knowledge acquisition are no longer valid but cannot easily be changed without redoing a major part of the knowledge-acquisition process.

For COMPASS, the No. 2 EAX domain that we selected was very stable. Through the entire development, no rule was ever altered because of a change in the No. 2 EAX architecture or control software.

Getting Started

Before discussing the major techniques for acquiring, recording, and documenting the expert knowledge (see the next two sections), let us consider some points related to getting started: how to set up the knowledge-acquisition meetings, what the first knowledge-acquisition meetings should cover, and what knowledge-acquisition techniques can be used at these initial meetings.

Knowledge Acquisition Meetings

The planning and scheduling of knowledge-acquisition meetings are important practical concerns.

• Organize knowledge-acquisition meetings so as to maximize access to the expert and to minimize interruptions.

As mentioned, the best experts often are the ones who can be least spared from their usual position. If an expert is consulted frequently for major and minor crises, knowledge-acquisition meetings held near the expert's location are likely to have many interruptions. It might be desirable to hold the meetings at a site remote from the expert's place of business. However, knowledge acquisition at the expert's site might allow observing the expert performing tasks in a usual environment, and this experience can be advantageous (see Starting Knowledge Acquisition).

We held our COMPASS knowledge-acquisition meetings at our Waltham, Massachusetts site, and our expert flew from San Angelo, Texas, to attend them. By having scheduled meetings in Massachusetts, we minimized--but did not completely eliminate--the times when our expert was called upon to help in crises, necessitating a rescheduling of our knowledge-acquisition meetings. However, once the expert was in Massachusetts, we could count on his availability except for occasional telephone calls.

• Knowledge-acquisition meeting attendees should have access to the implementation machines.

Several reasons exist for running the developing expert system program during a knowledge-acquisition session: to check parts of the developing program, to examine results of new knowledge that was acquired and implemented during the session, and to use the output of a part of the program as test input for knowledge acquisition of a succeeding part of the program. Thus, it is important to have access to the implementation during knowledge acquisition. This access can be achieved by having the knowledge-acquisition meetings at the location of the knowledge engineering team. Meetings at this location might offer additional benefits, such as decreased travel expenses and better access to knowledge-acquisition aids, but it might also increase interruptions for other business.

Having COMPASS knowledge-acquisition meetings at our Waltham site gave us immediate access to the COMPASS program for the purposes described. We also minimized travel expenses because one expert, rather than two to four knowledge engineers, had to travel. Work at our site facilitated use of our knowledge-recording and documentation-updating mechanisms. A negative aspect was that the knowledge engineers were sometimes called away to attend various meetings, delaying knowledge acquisition. However, because our primary job was developing COMPASS (as opposed to the expert's primary job: his work at GTSW), we were able to schedule other meetings so that we were rarely called away for long periods.

Getting Background Domain Knowledge

The knowledge engineers developing an expert system are often completely unfamiliar with the domain of the system. Thus, as part of the knowledge acquisition, they must be provided with some background in the domain.

• An initial period of the knowledge acquisition should be devoted to the expert giving the knowledge engineers a tutorial on the domain and the domain terminology, without any actual knowledge acquisition going on.

Although there is a natural impatience to get right into the "real" knowledge acquisition, domain concepts and terms will occur over and over in the knowledge-acquisition meetings. Thus, it is useful to invest some time up front discussing the domain in general without focusing on the specific task to be performed by the system.

In COMPASS, we devoted the entire first week of knowledge-acquisition meetings with our expert to a tutorial on telephone switching in general and on the No. 2 EAX structure. During this week, no mention was made of the specific task of COMPASS--the analysis of maintenance messages. Instead, the knowledge engineers learned a lot of basic telephone-switching ideas and No. 2 EAX jargon that would prove very useful during the remainder of the knowledge acquisition.

• Preparation of a tutorial document on the domain is useful.

This document can be used during the initial tutorial period and can then be available to knowledge engineers who join the project at later stages.

In COMPASS, the expert prepared a tutorial document that consisted of a package of pertinent excerpts of several existing No. 2 EAX reports and publications. This document provided the knowledge engineers with a useful reference during and after the tutorial week. Also, a copy of the document was given to each of the three new project members who eventually joined the COMPASS project. The expert also gave the new individuals private minitutorials as needed, based on the document.

Starting Knowledge Acquisition

Once the knowledge engineers have some basic background in the domain, it is time to start the actual knowledge acquisition.

• References such as books or other written materials discussing the domain can form the basis of an initial knowledge base.

In a book, an expert has already extracted and organized some of the domain expertise. This organized knowledge might prove useful (at least initially) in building the system.

As mentioned, in COMPASS, we used the existing No. 2 EAX reports and publications to help us gain general knowledge about the No. 2 EAX. However, no written materials explained the kind of analysis our expert went through to find and repair problems in the switch; therefore, we could not get any initial rules directly from books or reports.

• Begin the knowledge acquisition by having the expert go through the task, explaining each step in detail.

If possible, the expert should slowly work through the task for some test cases, explaining each step in detail. This task, however, is usually very difficult for the expert. An alternative is to have the expert perform the task at close to normal speed, verbalizing whenever possible, and record the process on audio or videotape. Another alternative is to record the expert on location the expert actually performs the task. In either of these cases, the tape of the task performance can be played back one short segment at a time, with the knowledge engineers attempting to find out from the expert exactly what is being considered and what decisions are being made at each point. A briefly considered decision by the expert often actually involves a very large amount of information that must be put into the expert system.

We initially used audiotapes in COMPASS, with our expert going through the task at close to normal speed. We obtained some initial idea of his domain techniques using this method. However, after a brief time, it became clear that our expert was able to slowly step through his analysis while we interrupted him at each step to probe for his methodology. Thus, we stopped taping after just a few days and relied upon this alternative.

NR 20 XY ANALYSIS RULES

BC DUAL EXPANSION ONE PGA DOMINANT LARGE NUMBER MESSAGES ANALYSIS RULE

IF There exists a BC Dual Expansion One PGA Dominant Problem
    AND
        The number of messages is five or more
THEN
        The fault is in the PGA in the indicated expansion (.5)
    AND
        The fault is in the PGA in the silent expansion (.3)
    AND
        The fault is in the IGA (.1)
    AND
        The fault is in the Backplane (.1)
BECAUSE
    Most messages are in one expansion, so the problem is probably in that PGA.

Figure 1. A COMPASS Rule.

Documenting the Knowledge

In order to fully discuss the techniques that can be used to elicit the knowledge, it is important to describe first what this knowledge will look like and what techniques can be used to document it.

• Use some form of quasi-English if-then rules to document expert knowledge whenever possible; use quasi-English procedures when rules cannot reasonably be used.

Utilizing if-then rules for documenting the knowledge acquisition allows the knowledge to be acquired in independent chunks, in a way that might become a basis for implementation. An expert should be able to understand this method of knowledge representation more easily than other AI paradigms and after some exposure might be able to relate some of the knowledge to the knowledge engineers by utilizing this paradigm. Other experts should be able to read and understand the documentation in this form for verification or technology-transfer purposes.

In the COMPASS knowledge acquisition, we used quasi-English if-then rules almost always. Occasionally, other forms of knowledge documentation were used. For example, when a complicated looping procedure was found, a procedure in English was documented. Additionally, when a large amount of data was found related to some items, data in tabular form were utilized. Our expert could easily read and refer to the rules. Later in the development, other No. 2 EAX experts were asked to evaluate COMPASS. They were able to read the rules and procedures in the knowledge document and understand the knowledge inside the system. Also, we were often able to implement a knowledge-acquisition rule by one or more implementation rules in KEE (usually with associated Lisp functions). When this implementation could be done, it allowed a nice isomorphism between the knowledge and the implementation that would not have been possible if the documented knowledge were not in rule form.

• Keep rules and procedures in a "knowledge document."

As the rules and procedures are found, keep them in a knowledge document. The knowledge engineers and the expert will be using the knowledge document frequently. It can be given to other experts for system verification purposes. It can also be considered a "specification" for the knowledge implementation. Finally, the knowledge document should become part of the final documentation of the project (possibly part of, or generated from, the corresponding implementation.) The COMPASS knowledge documents (Prerau et al. 1985b, 1986), were used extensively throughout the project for all these purposes.

• Develop conventions for documenting the knowledge-acquisition rules in order to add clarity.

Because the knowledge document can be used for many purposes, clarity is important.

A rule from the COMPASS system documentation (Prerau et al. 1986) is shown in figure 1. Note the use of capitalization and indentation to make the rule readable. The four points that follow highlight some of the other conventions we used.

• The rules and procedures in the knowledge document should use standard domain jargon that the expert and other domain practitioners can understand; any special conventions should be clearly specified in the document.

This practice allows the document to be used for all the purposes mentioned.

The COMPASS rule shown in figure 1 uses No. 2 EAX domain jargon (for example, "PGA" and "expansion"). Also, the COMPASS knowledge document makes clear (though this point is not evident in the figure) that a number in parentheses in the rule represents the likelihood that the fault is in the cited location.

• Group the documented rules in reasonable divisions.

Organizing the rules aids the user in finding them in the document. It also puts related rules together, which often facilitates document editing. If possible, the implementation should follow this grouping, but grouping the documented rules is a useful procedure to follow even if the implementation cannot correspond to the grouping.

In figure 1, NR 20 XY ANALYSIS RULES refers to a set of analysis rules (Analysis is one major phase of COMPASS) for network recovery 20 messages that deal with problems of the switch which the expert would group under the term "XY".

• Give the knowledge-acquisition rules unique descriptive names (lengthy if necessary) rather than numbers.

The rule name should be descriptive enough to ensure that it is unique. If a rule name is descriptive, it can be utilized as part of the explanation facility of the expert system. If possible, it can also be used to identify the corresponding part of the implementation. Rule numbering should be avoided because of the problems it can cause. The set of rules changes continually during knowledge acquisition (and later in program maintenance). Rules are regrouped; new rules are added; and old rules are deleted, combined, or split. If rules are numbered, they must be renumbered continually. Lengthy names are cumbersome, but they clearly define the rule and remain constant as the rule set changes.

In figure 1, the COMPASS rule name BC DUAL EXPANSION ONE PGA DOMINANT LARGE NUMBER MESSAGES ANALYSIS RULE identifies the expert rule to be applied in the analysis phase of the task under the following specific situation: the system has narrowed the problem to the BC portion of the switch, there are two expansions, the number of messages for one of the two PGAs is significantly more than (dominates) the number of messages for the second PGA, and the total number of messages is large (defined in this rule as "five or more"). The very long rule name can stay with the rule no matter how other rules change. In the COMPASS implementation, the KEE rule that implements this knowledge rule is given the same name (within allowable rule-name syntax).

• Include an explanatory clause as a part of each rule.

An explanatory clause (for example, a BECAUSE clause) appended to an if-then rule provides additional information on the expert's justification for a rule (Kyle 1985). Although this clause has no effect on the operation of the expert system, it can help the expert and the knowledge-acquisition team remember why they defined certain rules as they did and can clarify these decisions for other experts and the maintainers of the system. The explanatory clause might also be used in a justification part of the system.

The COMPASS rules originally did not have an explanatory clause. We found that occasionally when we examined a knowledge rule which we hadn't looked at in a while, we could not remember exactly why something was done one way rather than another. We wasted valuable time reconstructing the arguments that were used previously. Furthermore, we found related problems occurred when the COMPASS rules were read by persons outside the COMPASS development team (for example, an outside expert who is examining the COMPASS knowledge base or someone involved with COMPASS maintenance). These people sometimes found it difficult to understand the reasoning behind certain parts of COMPASS. The addition of BECAUSE clauses to COMPASS knowledge rules that were not self-explanatory minimized these problems and should help reduce future problems.

Acquiring and Recording the Knowledge

The major work in the knowledge-acquisition process is the lengthy time spent with an expert eliciting, modifying, and recording the domain knowledge.

Acquiring the Knowledge

• Follow a basic cycle of elicit, document (or document and implement), and test.

An effective method of knowledge acquisition is to use the following basic cycle: (1) elicit knowledge from the expert; (2) document and, if possible, implement the knowledge; and (3) test the knowledge by comparing the expert's analysis against hand simulations of the documented knowledge or against the implementation (see figures 2 and 3).

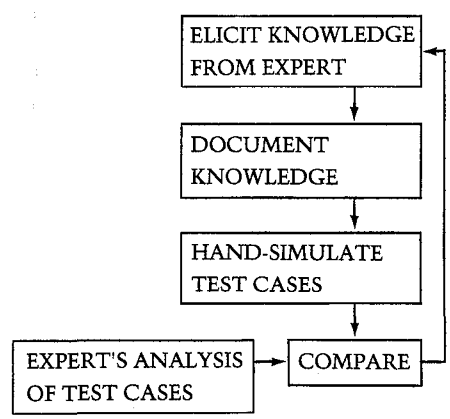


Figure 2. Knowledge-Acquisition Cycle with Hand Simulation.

We used this method for the COMPASS knowledge acquisition. We found hand simulations best for examining each small step of reasoning, while comparisons against the implementation were most useful when a large section of knowledge had been

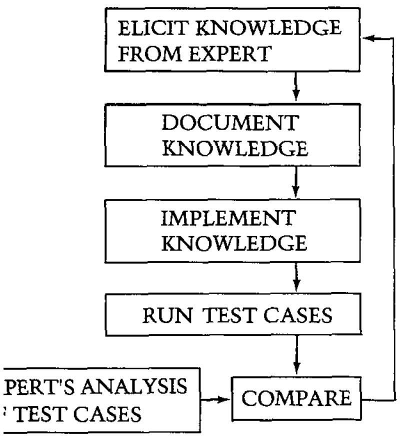


Figure 3. Knowledge-Acquisition Cycle with Program Runs.

completed.

• Use test cases to elicit expert techniques.

When initially considering a new area of the domain, go through several test cases. For each test case, formulate and document the rules and procedures the expert used to perform the task.

We went through each test case with the COMPASS expert. He tried to explain each substep in as much detail as he could, and we formulated knowledge-acquisition if-then rules or procedures to document each substep. We discussed each rule with the expert and modified it until he was satisfied. Because he knew each rule was to be considered just an initial version of the rule and would be subject to much change in the future, he did not feel that accepting a rule was a major decision requiring a great deal of thought.

• Use a large number of additional test cases to expand and modify the initial knowledge.

Go through numerous additional test cases. For each test case, attempt to use the existing rules and procedures to perform the task for the test case. Do this process by hand, or, if the pertinent rules and procedures have already been implemented, by machine. In each case, have the expert examine the reasoning of the system step by step. Find all points of disagreement between the expert and the system, and modify and expand the existing rules and procedures so that they work correctly, that is, they agree with the expert. As the system gets bigger and the implementation grows, you can compare the expert's final results with those of the system and examine in detail only those cases where there is disagreement.

We went through numerous test cases with the COMPASS expert. Rules and procedures were continually changed. At some points fairly early in the process, we (including the expert) thought that we were almost finished. Subsequently, we would find test cases that opened up completely new areas which hadn't been considered, and we would find other test cases that pointed to major required changes and expansions to the existing rules and procedures. The expert made many subtle decisions and checks in his analysis that he did not realize he was making until a test case pointed them out.

• Have the expert define the domain reasoning in terms of knowledge-acquisition rules and procedures if possible.

Explain to the expert the ways the knowledge will be documented. As the knowledge-acquisition process continues, an interested expert will begin to understand the use of if-then rules and other AI concepts (just as the knowledge engineers will begin to understand some of the deeper concepts of the expert's domain). This understanding might help the expert describe the domain knowledge by directly using the knowledge-acquisition formalisms, thus speeding the knowledge-acquisition process. In addition, it helps the expert interpret the knowledge base being built and provides a foundation for the expert to eventually participate in the maintenance of the expert system implementation.

As the knowledge-acquisition process continued, the COMPASS expert became increasingly familiar with the rule formalism we used and often was able to formulate his domain expertise in this form. (At the same time, the COMPASS knowledge engineers slowly became No. 2 EAX miniexperts.)

• As knowledge is acquired and updated, generate and continually update the knowledge document.

During the knowledge-acquisition sessions, each knowledge engineer and expert should have an up-to-date copy of the knowledge document. When knowledge is being acquired and modified rapidly, new versions of the document should be printed as soon as possible.

We had our COMPASS knowledge document in our word processor. We updated and reprinted it after every one-week knowledge-acquisition session--at the least. When knowledge acquisition was rapid, we updated and reprinted it daily or even more frequently. (The COMPASS knowledge document was time-stamped to the nearest second--only a slight exaggeration of what was required.)

• Be general in wording, if necessary, when initially defining knowledge-acquisition rules and procedures.

Use general phrases in the rules each time the expert has trouble detailing or quantifying a specific knowledge item. This procedure avoids getting the knowledge-acquisition session bogged down in minor details before the important problems are solved. Later in the knowledge-acquisition process, the general phrase can be replaced by a specific quantity if possible, or techniques for dealing with uncertainty can be used.

During COMPASS development we used this technique several times. For example, the phrase "a sufficient number of messages" was used as a part of several rules for several months. A rule might state, "IF X is true AND there is a sufficient number of messages, THEN conclude Y." The phrase was given a working definition (=5) for a time to allow initial rule implementation and only after several months was the phrase replaced in the rules by a specific number. The number turned out to be different for different rules.

• Use each test case to generate many additional test cases.

When a test case has X = 5 and a rule is formulated, ask the expert whether the rule would be the same if X were 1, X were 10, and so on. By going through this process in the middle of the discussion of the original test case, the entire context of the test case does not have to be rediscussed in order to come to the point at issue.

Several new knowledge rules are often quickly generated in this manner

This technique worked very well in COMPASS, frequently allowing us to examine several different situations based on a single test case.

• Use the generated test cases to find the "edges" of each rule.

If a rule applies for X = 10 and another applies for X = 20, ask the expert which rule applies if X is 15, 17, and so on. Such questioning might make the expert uncomfortable because the rules of thumb often do not have sharp boundaries. However, after some thought, the expert might be able to pick a reasonable cutoff point. Note that if the expert is unsure which of two rules applies for a certain situation, the expert system might not be too far wrong if it uses either one; so, the expert's selection might not be critical. Again, this technique worked very well in COMPASS.

Recording the Knowledge

The final documentation of the knowledge was previously discussed. Here, let us consider techniques for initially recording the knowledge as it is acquired during the knowledge-acquisition sessions.

• Record acquired knowledge in a flexible manner at the knowledge-acquisition sessions.

The method in which the acquired knowledge is initially recorded at the knowledge-acquisition sessions should allow for frequent changes in rules while they are being discussed It also should facilitate the transfer of the knowledge to the knowledge document when the discussion is completed. It would be efficient if the initial knowledge recording could be immediately and automatically transformed into the knowledge documentation (and even better if the documentation could then be transformed into the implementation). However, if this transformation cannot take place easily, it is wise to use the recording technique best suited to the knowledge acquisition rather than to delay the knowledge-acquisition sessions while the documentation and implementation are being produced.

In the COMPASS knowledge-acquisition sessions, we wrote the knowledge-acquisition rules and procedures on a whiteboard, and after a rule or procedure had been agreed on, we took an instant photograph of the board. The knowledge document was updated as soon as possible after a day's knowledge-acquisition session using the photographs (which were kept on file for reference) This technique proved very useful in COMPASS, but it did require that a project member spend a significant amount of time transcribing the information from the photographs.

• Use suitable conventions for knowledge recording.

To speed the knowledge recording process, develop some reasonable recording conventions

In COMPASS, we found that adopting a color code for different categories of information (for example, new rules, revisions of old rules, comments, and so on) was initially a help After several months as we became familiar with our knowledge-recording and -transcribing process, we abandoned the color coding.

• Use reminders to defer overly detailed or secondary items

During knowledge acquisition, you sometimes come upon topics to discuss or actions to take that are beyond the scope of the current discussion. For example, one obscure case might be complicated enough to require a significant amount of knowledge acquisition. Rather than diverting the knowledge-acquisition session into a very detailed area or, alternatively, neglecting the topic, it is useful to have a formal mechanism to record "reminders" that can trigger a knowledge-acquisition session at a later date.

In the COMPASS knowledge-acquisition meetings, reminders were treated as an outcome of knowledge acquisition, similar to the rules and procedures. These reminders were updated, deleted, or added to in the same manner as the other knowledge. Every so often, the group would go through the reminder list to see if any reminder should be treated immediately or if any could be deleted as no longer necessary.

A Related Implementation Convention

• Implement knowledge acquisition rules by a corresponding implementation rule(s) with the same name if possible.

Using the same name for a knowledge-acquisition rule and its corresponding rule(s) in the implementation helps keep the knowledge acquirers from worrying about the details of the implementation, yet it preserves the correspondence between the acquired rules and the implemented rules It greatly aids technology transfer and maintenance because it makes a clear correspondence between the knowledge document and the implementation Use of this technique also facilitates implementation.

As mentioned earlier, COMPASS is implemented using multiple AI paradigms (rules, frames, demons, object-oriented programming, and Lisp), making use of all the facilities of KEE However, to maximize the maintainability of COMPASS, we tried to use KEE rules to implement knowledge-acquisition rules whenever possible. Furthermore, when possible, we tried to have a one-to-one correspondence between the knowledge rules and the KEE rules. Although it might have been more efficient in many cases to use a different paradigm (for example, Lisp code), we feel that the use of KEE rules to implement knowledge-acquisition rules makes COMPASS much easier to understand and maintain, which was a priority for us (Prerau et al., 1987). Each KEE rule used was given the same name as the corresponding knowledge-acquisition rule. When a knowledge-acquisition rule was implemented by multiple KEE rules, then the implementation rules were given the same name as the knowledge rule but with a number added Thus, the COMPASS knowledge rule F-SWITCH ANALYSIS RULE is implemented by two KEE implementation rules named F-SWITCH ANALYSIS RULE #1 and F-SWITCH ANALYSIS RULE #2.

Conclusions

This article presented several effective techniques for knowledge acquisition and some of the details of a real expert system knowledge-acquisition process. It is doubtful that every point discussed will be usable or even pertinent to the knowledge-acquisition task for another project. However, until some general theories of knowledge acquisition become accepted, it is important for expert system developers to describe successful techniques they have used in order to allow others to glean what they can. Some of the principal techniques that were found to be beneficial in knowledge acquisition are: (1) considering knowledge acquisition when selecting the domain, (2) considering knowledge acquisition when selecting an expert, (3) using test cases to elicit knowledge, (4) using generated test cases to multiply the effectiveness of test-case analysis, and (5) using good knowledge-recording and -documentation practices.

Acknowledgments

I would like to thank Dr Shri Goyal and members of the Knowledge-Based Systems Department at GTE Laboratories for their continuing support in this work I especially acknowledge the excellent contributions of Dr Alan Lemmon, Robert Reinke, Alan Gunderson, and Dr Mark Adler to the COMPASS knowledge acquisition and the development of the knowledge-acquisition techniques described in this article. I'd like to thank Scott Schipper of GTE Data Services for his contribution to the knowledge acquisition and also his initial documentation of some of the COMPASS knowledge-acquisition techniques. Scott's work was a catalyst for this article. The knowledge acquisition would not have been successful without the fine job done by our expert, W. (Rick) Johnson of General Telephone of the Southwest. Finally, I'd like to thank Dr. Charles Rich of the Massachusetts Institute of Technology, who provided many useful suggestions throughout the course of the work that led to this article.

References

Fikes, R., and Kehler, T. 1985. The Role of Frame-Based Representation in Reasoning. *Communications of the ACM* 28(9): 904-920.

Goyal, S.; Prerau, D. S.; Lemmon, A.; Gunderson, A.; and Reinke, R 1985 COMPASS: An Expert System for Telephone Switch Maintenance. *Expert Systems: The International Journal of Knowledge Engineering* 2(3): 112-126. Also published in Proceedings of the Expert Systems in Government Symposium, 112-122. New York: Institute of Electrical and Electronics Engineers.

Kyle, T 1985. Expanding Expertise by Use of an Expert System. In Proceedings of the 1985 Conference on Intelligent Systems and Machines, 244-247 Rochester, Mich.: Oakland University

Prerau, D. S 1985. Selection of an Appropriate Domain for an Expert System *AI Magazine* 4(2): 26-30.

Prerau, D. S ; Gunderson, A.; Reinke, R ; and Goyal, S. 1985a. The COMPASS Expert System: Verification, Technology Transfer, and Expansion. In Proceedings of the Second International Conference of Artificial Intelligence Applications, 597-602 Washington D.C.: IEEE Computer Society

Prerau, D. S ; Lemmon, A ; Gunderson, A.; and Reinke, R 1985b. A Multi-Paradigm Expert System for Maintenance of an Electronic Telephone Exchange. In Proceedings of the Ninth International Computer Software and Applications Conference (COMPSAC-85), 280-286 Washington D C.: IEEE Computer Society

Prerau, D S.; Lemmon, A.; Gunderson, A.; Reinke, R.; and Johnson, W. 1985c. COMPASS-X Rules and Procedures, Technical Note, TN-85-176.2, Computer Science Laboratory, GTE Laboratories Incorporated.

Prerau, D S.; Schipper, S.; and Janis, J 1985d Artificial Intelligence Technology Transfer Related to the COMPASS-X Expert System, Technical Note, TN-85-176.6, Computer Science Laboratory, GTE Laboratories Incorporated.

Prerau, D. S.; Gunderson, A.; Reinke, R ; Adler, M ; Johnson, W.; and Schipper, S. 1986. COMPASS-II Knowledge Document, Vols. I to IV, Technical Notes, TN86-176.8 to TN86-176.11, Computer and Intelligent Systems Laboratory, GTE Laboratories Incorporated.

Prerau, D. S.; Gunderson, A.; Reinke, R.; Adler, M. 1987. Maintainability Techniques in Developing Large Expert Systems. Forthcoming.

An extended version of this article will appear as a chapter in a forthcoming book by Dr. Prerau. This book discusses all aspects of the development of expert systems based on direct experience in large expert system projects in industry.