KNOWLEDGE ENGINEERING:

THEORY AND PRACTICE

H. Joel Jeffrey

ABSTRACT

The most difficult part of building an expert system (one that models significant human expertise) is knowledge engineering, the art of gathering expert human knowledge and representing it in technically usable form. Since the knowledge engineer's goal is complete, precise, technically usable representations of human behavior, and Descriptive Psychology is a systematic formulation of the concepts of person, behavior, language, and the real world, one would expect Descriptive Psychology to be very useful in knowledge engineering, and this has proven to be the case. In the last several years considerable experience has been gained in using the formulations of Descriptive Psychology to do knowledge engineering in a variety of areas. This paper presents some of these formulations, and the concepts, approaches, and practices based on them.

The past ten years has seen the emergence of an area of computer science and technology known as "expert systems". An expert system is one which attempts to reproduce the behavior of a human expert or

Advances in Descriptive Psychology, Volume 5, pages 123-146.
Editors: Anthony O. Putman and Keith E. Davis.

ISBN: 0-9625661-0-1.

experts in some domain. It is widely agreed that the most difficult part of building an expert system is getting the expertise from the human and representing it in a form that the computer can use it. That enterprise is known as knowledge engineering.

The author has been engaged (with others, notably A. O. Putman and M. E. Haarer) for several years in building expert systems. We have successfully produced several systems, covering a wide range of expertise, which compare favorably in size with other expert systems. (One of these, MENTOR, is discussed in detail in Jeffrey & Putman, 1983.)

In the course of this work, we have used, and in some cases developed, a number of concepts and practices which we have found make a substantial difference in our ability to gather the knowledge necessary and represent it in computer-useable form. This paper presents these approaches, concepts, and practices.

The paper discusses the logic of representing human behavior, and the practice of knowledge engineering as a human enterprise. It addresses the twin issues of what one must specify in order to represent certain types of human expertise, and how one goes about gathering the necessary information. The logical requirements that any form of representation must meet in order to qualify as a description of this type of expertise are presented, and it is shown why certain technical developments in artificial intelligence (e.g., "frames" and "frame systems") have proven so attractive to workers in the expert system area, but so difficult to use effectively.

This paper is intended to address (at least) two distinct audiences: Descriptive Psychologists interested in applications of Descriptive Psychology in technical areas and/or other practical realms outside clinical psychology, and those familiar with expert systems who are interested in a different approach to the problems encountered in building them. Descriptive Psychologists may find the development of the parameters of Intentional Action and Social Practice Descriptions to be a review of familiar material. Others may face more a difficult problem. The discussion of human action presented in this paper is deliberately couched in common terminology, but includes a number of highly technical concepts, such as intentional action, social practice, knowledge, perspective, and skill, to name a few. Keeping this in mind may help to avoid the impression that the discussion is "loose" or informal. The interested reader is referred to Ossorio (1970/1981), Jeffrey and Putman (1983), and Putman (1981) for more detailed presentation of these concepts and their relation to other fields of psychology.

The approach to expert systems, and artificial intelligence in general, exemplified in this paper differs in two significant ways from the more usual ones. First, the central concept is action, rather than knowledge. Expert behavior in a domain is reproduced by representing, at a useful level of detail, everything it takes to engage in the behavior (or behaviors), and having a software system that can act on those descriptions. Knowledge is one (but only one, as we shall see below) of the necessary ingredients of behavior descriptions. By contrast, the more usual approach is to treat a person as an "information processing mechanism", and actions as logical implications of the information (Firschein, 1984).

The second major difference is to view expert behavior as a case of human behavior, and to note that knowledge engineering is a case of one person describing another's behavior. It is therefore a human enterprise, in which the relevant concepts, skills, and perspectives (about which much more will be said below) are those which are oriented to understanding and describing another human's behavior, rather than those from mathematics, computer science, or any other technical realm.

BACKGROUND

Ever since computers began to be commonly available in approximately the early 1950s, researchers have attempted to program computers to do things that, in the common idiom, are called "intelligent". By this it is meant that the things are not done merely "by rote", but require analysis, judgement, skill, or some combination thereof. This field has come to be called "artificial intelligence".

Artificial intelligence includes several distinct areas. Examples include programs to play games "intelligently" (i.e., not simply by a procedure with a guaranteed result); programs to solve problems; and programs to understand natural language.

In recent years the area of expert systems has become quite prominent, bearing the fruit of actual successes and programs of practical use outside the academic community. As human expertise is expensive and rare, there is great demand for such systems. The core practice in this area is the production of computer programs that reproduce the behavior of some human expert or experts (Rich, 1983).

Examples of working expert systems include MYCIN, a medical diagnosis system (Buchanan & Shortliffe, 1984); PROSPECTOR, a geological data analysis program (Duda & Reboh, 1984); and R1/XCON, a program to configure computer systems (Kraft, 1984). An expert system, then, is a computer program that engages with a person as a human expert in some area does. Another way of saying this is that it makes sense to view the program as engaging in the social practices that the human expert does. (This does not mean that the human expertise has been reduced to an algorithm. This issue has been addressed in some detail in Jeffrey, 1981.)

It is the knowledge engineer's job to capture and represent what the expert does, in such a way that the system can engage in the practices of interest.

Obviously, the form available to the knowledge engineer for representing the expertise will heavily influence representation. Perhaps not so obviously, this form may, and often does, also influence what expertise the knowledge engineer gathers and how he gathers it. Further, the expertise (and all of its aspects) is not the same as the form for representing it; representing human expertise requires that we have a statement of what that expertise consists of, independent of the form for representing it.

THE INFORMATION TO BE GATHERED

"Knowledge Engineering" vs. "Action Representation"

The term "knowledge engineering" is standard terminology in the expert system field. It reflects a certain approach to the problem of reproducing human expertise. This approach is to consider a person's knowledge a "thing" which he has, and which he "applies" to a problem, the outcome being some performance. With this approach, the emphasis is on finding out what the expert "knows" and then connecting this knowledge to the performance. In most systems, the knowledge is connected to the actions by "rules" of tbe form, "If X, then Y". The particular Xs and Ys are found by talking to the expert, or experts, and may be either a further item of knowledge or a performance. Here is an example, from the MYCIN system:

> If: the stain of the organism is gram-positive, and the morphology of the organism is coccus, and the growth conformation of the organism is clumps
>
> Then: (with certainty 0.7) the identity of the organism is staphylococcus (Rich, 1983, p. 286)

I believe that this term, although standard, is something of a misnomer. Taken literally, it indicates that one is building some object, or construct, out of "knowledge". Further, the knowledge engineer almost inevitably focuses one on the knowledge aspect of tbe enterprise. However, what matters is whether the system reproduces tbe "performances" (using the term as it is used outside Descriptive

Psychology) of the expert, not the knowledge itself. In other words, it is the behavior of the expert that is to be reproduced, and I believe therefore that a more appropriate focus, and terminology, is *action representation.*

The action representation approach to the task of producing an expert system is not to ask, "What does the expert know that allows him to perform in this way", but rather, "Exactly what does this expert do?", or, more technically, "What are the Social Practices that this person engages in?"

To answer this question, the knowledge engineer presents social practice descriptions of the intentional actions the person engages in. These social practice descriptions are described in detail below.

Approaching the task in this way does not eliminate the need to specify the expert's knowledge. Rather, it expands the task of the expert system builder to specify not only the knowledge, but all of the other aspects of the actions the expert engages in, and their relationships.

The following discussion addresses the parameters of intentional action and how one acquires the information that allows one to fill in the parameters in actual cases. The presentation begins with an actual case and develops each "type of knowledge" (i.e., parameter) necessary.

This form of presentation is intended to highlight, and emphasize, the close connection between ordinary "common sense" understanding of human actions and the technical ways of acting on that understanding that one must have for technical work.

A further point is in order. As has been discussed at length elsewhere (Ossorio, 1970/1981), Descriptive Psychology is not simply another theory or approach; it is the systematic articulation of the concepts of person, behavior, language, and the real world. This contrasts with a theory, abstraction, construct, etc. The particular form used to represent the aspects of a person's actions is open to a great deal of personal preference and choice; the logical requirements any such representation format must meet are not. As a result, using the formulations of Descriptive Psychology to understand, and represent, a person's actions (i.e., knowledge engineering) contrasts, in some cases quite sharply, with using abstractions such as frames, if-then rules, or mathematical logic (Winston, 1984). It is hoped that this form of presentation will make these differences clearer to the reader.

What Kinds of Information are Needed

Let us begin by presenting a some actual data from an expert. The types of information present in this data will then be discussed. After this analysis, other types of information (aspects of a description of a social practice) that are not represented directly in this data, but which we can recognize as necessary for a complete picture, will be discussed.

This data is a verbatim fragment of an interview with a computer programmer of many years experience. The interviewer asked the person to explain how he locates and identifies errors in a computer program, based on observed erroneous performance by the program (known colloquially as "fixing a bug").

This topic is presented here because it is widely accepted as a practice requiring considerable skill and analysis, as well as certain observable Performances. In other words, to describe "bug fixing" one must address several aspects of intentional action, both the "hard" and "soft" aspects (as they are commonly called, albeit not by Descriptive Psychologists).

This interview also provides excellent examples of the sort of language one encounters in actual interviews, and thus is good raw material for the later section of this paper, in which some of the practical issues of interviewing people and understanding what they say are addressed.

> Finding a bug in a program is a job of eliminating all the places where the bug isn't. Anything you can do to shrink the possibilities is a step in the right direction.
>
> Sometimes I will first just run the program a half a dozen times to be sure I get the feel of it—what it's supposed to do. But the thing I have to be able to do, before anything else can begin, is to reproduce the bug.
>
> Once I can reproduce it, I follow two rules of thumb. First, do anything that will narrow the search, and second, do the easy stuff first. Experience shows that doing the easy tests first is often helpful even if you don't think the bug is in the areas you can easily test.
>
> You have to have a mental image of what the program is supposed to do. One way to find out where the program and your mental image are out of harmony is to add code to the program. This lets you test what the program does against what you expect it to do.

Process Information

Perhaps most obvious type of information in this interview is procedural; actions that a programmer takes in order to "find a bug". These include "reproducing the problem", "narrowing the range where the problem could lie", "getting an understanding of what the program is supposed to do", "adding code to the program", and "running a test".

Each of these actions are part of the overall practice of fixing a problem in a computer program; they are steps a programmer takes, or may take, in order to carry out the practice. They may not all be taken, and may not be taken in the order listed here or the order mentioned by the expert interviewed; they constitute a list, with no order implied, of the tasks involved in this practice. These are the *stages* (using now the technical Descriptive Psychology term) of this practice.

Next, some of the stages we have listed have associated with them certain rules for when they are done. These rules, or constraints, may be giving a certain order to the stages ("the first thing to do is . . .") or to the circumstances under which they are done ("if I don't have a good feel for the what the program is supposed to do . . ."). These rules are the *attributional* and *co-occurrence constraints* of a process (Jeffrey & Putman, 1983; Ossorio, 1971/1978).

These Constraints, and the expert's report, give the possible sequences of the Stages that one might encounter in an actual occasion of a programmer finding a bug. These sequences (technically, the *versions*) of the practice being described complete description of the procedural, or Performance, aspect of the description.

Readers familiar with rule-based expert systems (Rich, 1983) will recognize constraints as rules. Further, carrying out a Stage can also be represented as a rule. The point is not that this formulation replaces rules (although it could, and has, in the systems we have built); but simply that there is a significant difference between the two types of rules, which is being noted here.

Knowledge

A good deal of the expert's discussion above is devoted to such apparently nebulous notions as "getting a feel for the program" and "a mental image of the program". One possible way to deal with this type of report is to invent categories of information, with these labels, and place this portion of the interview information literally in these categories. This approach has been taken for example by Schank (Schank & Riesbeck, 1981), in which categories such as "mental transfer" are used to denote a person telling another person something. It is demonstrated in detail by Weilinga and Breuker (1985) and Ferrand (1985).

Our approach to this type of data is less literal but, it is hoped, logically tighter. When a person acts, he is acting on several items of knowledge—things he knows. (This is articulated in detail in Ossorio 1970/1981.) In discussing what he knows, a person may not (and in fact usually will not) use language that states directly that this is something he knows in doing this practice. There are many language constructs for expressing this distinction; a person will use the ones he prefers (perhaps for a variety of reasons). Our approach is to note that the expert informant, with whatever language he uses, is referring to some item of knowledge that matters in engaging in this practice, and will represent each such item explicitly in the description of the practice. (As will be seen below, the expert's language for these items is used in another way also.)

This is not "interpreting" the knowledge, or "filling in incomplete knowledge", the more usual approach (Weilinga & Breuker, 1985). It is recognizing the concept, fact, or perspective the person is acting on.

Thus, another type of information that the expert has given in the above interview is the knowledge he has; technically, the values for the *know* parameter of the action. The job of the person observing and describing the expert's actions (the knowledge engineer) is to identify exactly what this knowledge is, for the social practice of interest.

In prior formulations of intentional action there was some ambiguity as to precisely what knowledge must be specified in order to represent an action (Ossorio, 1970/1981). In particular, it has not been clear whether the knowledge required for a stage of the practice belongs in the description of the practice itself. If one had to include all items of knowledge from a stage, there would be significant technical difficulties, because stages are themselves social practices, and thus subject to further description, down to whatever level of detail is needed.

The necessary clarification is this: the knowledge one must specify in order to give a social practice description is exactly those things the person must know to engage in *this* practice (but not some stage of the practice) (Putman, 1985).

It is useful to distinguish three types of knowledge. First, the person must have the facts of the particular case. If the expert in the above interview is to debug some program, clearly he must know what program it is; if a manager is to interview a candidate for a job, he must know which candidate and which job.

Second, the person acting must have the relevant concepts—that is, be able to make the relevant distinctions. The programmer must be able to distinguish between proper and improper program performance; in several of the practices one finds in psycbotherapy, therapist must know the difference between authentic and inauthentic behavior (although he certainly need not need use these words for the distinction).

Finally, the person acting always views the situation from a certain perspective. The concept of perspective is elaborated in detail by Putman (1985). Briefly, each status of a community has its perspective, and one sees "the facts" from that perspective. Further, one values certain states of affairs over others, and chooses actions that reflect these values (Ossorio, 1981/1983). (This formulation of perspective differs substantially from the semantic net formulation, in which perspective is equated to purpose. The reader is referred to Winston, 1984, pp. 263-265, for more information on that usage.)

Often, adopting the appropriate perspective is necessary to successfully carry out the practice. For example, one of the practices involved in designing and building a computer system is interviewing prospective users of the system, to gain an understanding of what they do and how the system would fit into their work practices. To do this, the interviewer must be able to adopt the user's perspective. If he can not, he may not recognize values and choice principles (Ossorio, 1981/1983) that play a significant part in the users' work practices, and thus the interviewer's descriptions of use practices will be an unreliable guide in deciding what the system should do and how it should look to a user.

Identifying the necessary perspective for engaging in a practice is often one of the more difficult tasks facing the knowledge engineer, because very often (and paradigmatically) the expert simply adopts the necessary perspective, without recognizing or being able to report that he does so. (This will be discussed further in the section on practical interviewing techniques.)

There is a further complication here. It is crucial to keep in mind that the knowledge a person must have to engage in a practice must be "present when they are doing it". That is, the person must be acting on the appropriate distinctions, facts, and perspectives. This in turn implies that it is not enough for the expert system merely to have the knowledge stored, or for the knowledge engineer simply to identify the knowledge, because merely telling a person what concepts, facts, and perspectives he needs will virtually never suffice to get him to be aware of the facts, make the distinctions, or adopt the perspectives. The expert system must also contain a representation of the performance the system must engage in to get the person to be aware of the fact, make the distinction, or adopt the perspective in question.

Consider again the example of the computer system designer. Suppose we are producing an expert system to assist someone in designing a computer system.

Here is one way in which this expert system might use the concept of perspective in assisting the human designer:

> The user's perspective is crucial here. What do you think that perspective is?
>
> (User replies)
>
> OK, Be sure to consider that perspective when you are designing the system.

This interaction is not likely to be more than minimally helpful to the designer, because while it does remind him of the user's perspective and its importance, it does not help the designer in adopting the user's perspective.

Here is an illustration of another approach, which we have used frequently with good results:

> The user's perspective is crucial here. Think over what you know of how the XYZ Department handles these forms. What seems to be the part of the job the people care the most about?
>
> (User replies)
>
> OK. Is there a part of this job that the people in the department would delegate to someone else if they could?
>
> (User replies)
>
> OK, Bob. One last question. How do the people in XYZ feel about the importance of these forms to Amalgamated Assets? Do they believe it is really important to get them done quickly and accurately, or do they feel they are basically "pushing paper?"

None of these questions and reminders are important for the content of the answers; the key is that in answering these questions the system designer is (at least very probably) looking at the job of the XYZ department as the people in that department do. In other words, the system designer is being asked questions that will tend to get them to adopt the perspective of a member of the XYZ department.

There is one other significant difference between these two illustrations. The first is couched in the language of an observer of the action the designer is engaged in; it is the language one might use to talk *about* the practice. The second is couched in actor language—the language one might use to talk *with someone engaged in the practice*.

At this point some of the differences between the usual approaches to knowledge engineering and that in this paper are visible. The more customary way of handling expert knowledge is to begin with a self-report or an observation, represent it mathematically, analyze it into logical primitives, apply mathematical transformations to the resulting representation, and attempt to fill the slots of the frame (Brachman, 1979; Weilinga & Breuker, 1985). In contrast, the key activity involved in representing knowledge as presented here is not analyzing it, inferring or deducing other knowledge from it, categorizing it, or analyzing its action implications. Rather, one begins with the action, and represents the knowledge needed as simply that: necessary facts, concepts, or perspectives. The items of knowledge then are referred to in the constraints on the stages and options.

The reason for this way of proceeding is not that analysis is somehow undesirable, but rather to keep all such analysis in its proper logical relation to other aspects of action. Specifically, the expert, or the expert system, may have to do something (which may include analysis) to gain some knowledge it needs in order to act, but the key to determining what to do next (including whether to analyze) is the action (the social practice) being done.

The reader familiar with more traditional approaches to knowledge representation may find some similarity between Stage-Options and Knowledge, on the one hand, and procedural and declarative knowledge on the other (Harmon & King, 1985). There is some formal similarity, but the distinctions are quite different. Declarative knowledge is a fact which is simply asserted, whereas procedural knowledge is a procedure which produces the fact. Stage-options are the formal means for representing what the person does, whether finding out some item of knowledge or anything else; knowledge is what one must know to carry out that practice. Some of this knowledge may be "procedural", in that one must do something to arrive at it.

Skills

It is common for experts to report doing things for which there is no procedure—no "how" they do it. For example, our expert programmer reports that he will "do anything that will narrow the range". One appropriate name of this practice is, "Narrow the range of the places where the bug might be". When asked how to do that, he can give a number of actions that might be helpful, but if asked how he chooses, he cannot answer the question. This is a case in which there is an action, which could appropriately be termed "choosing a technique to narrow the range where the bug might be", which the expert simply *knows how* to do; there is no other "how". In other words, this is a skill a programmer engaged in this practice must have.

Descriptive Psychologists will recognize that this is by no means uncommon, and we have found it so in actual practice. An example from another arena is instructive: when interviewing a person for a job, there are certain things one can do that will make a significant difference in carrying out the practice, but are not procedural; there is no "how". These skills include getting a person talking openly and candidly; unobtrusively drawing a person out on a topic; and assessing whether a person has a personal characteristic, based on what they say and do in the interview.

It should be noted that while there may not be any Performance that constitutes carrying out this practice, this does not mean that there is nothing to say about it. Part of the knowledge engineer's task is to recognize when the expert is exercising a skill (and thus when there is no point in asking how they do this particular thing), and then skillfully finding out what the expert can say about it that is useful to someone else. (Again, more on this later in this paper.)

The distinction being drawn here is not that between declarative and procedural knowledge one may find in the cognition literature (Harmon & King, 1985). It is not a distinction between types of knowledge at all. Knowledge refers to concepts (distinctions), facts, and perspectives. The role it plays in an action, as we have seen above, is that without the various things one must know, one cannot choose between stages, or options of stages, for these are constrained (in reality) by states of affairs which the actor must properly appraise. These logical relations are represented by attributional and co-occurrence constraints on stages; the states of affairs in these constraints refer to items of knowledge in the social practice description. Skills, by contrast, refer to procedures (technically, Performances in the intentional action description) which the actor simply carries out, with no need of (and in fact no use for) a social practice description describing how to do it. Such skills are extremely common in human action, although particularly obvious in expert behavior. (The reader is referred to Ossorio, 1970/1981 for further discussion of skills.)

Elements and Eligibilities

In the above interview fragment, it is easy to observe that there are various objects involved: the programmer, the program, the bug, etc. Further, any particular instance of debugging a program will involve particulars varying from case to case, but the logical *elements* will remain the same: program, bug, etc. This category of information addresses the "object" aspects of the practice (Ossorio, 1971/1978).

There are three aspects of this information: the *elements*, *individuals*, which are the actual particulars of the case, and *eligibilities*, which are the logical rules stating which individuals may take the place of which elements. For example, the role of "bug" (the element) might be specified by the individual with the name "Failure Report 0016A", or "the problem Jane found on October 13".

Just as rules constrain stages, rules may constrain which particular individual is used for a given element. This is also part of the information the expert provides.

Paradigms

Sometimes one can recognize two or more ways a practice can occur which have very little relationship to one another, other than being in fact ways of engaging in this practice. This is discussed in some detail by Jeffrey and Putman (1983) and Ossorio (1971/1978). The information discussed so far (the stages, versions, elements, individuals, eligibilities, and constraints) comprises one paradigm. Any further paradigms consist of the same logical elements.

Significance

A very important part of the information needed to represent human expertise is part-whole information: the larger social practice this practice is part of (if any).

For example, debugging a program might be a stage of developing a new program or making a change to an existing one. Developing a program, in turn, might be a stage of developing a new software product, making an addition to an existing product, producing a software tool to be used inside the organization, or experimenting with a new approach to a problem (to name a few).

It is quite common for a person to act differently—that is, do "the same thing" differently—when that action is part of different social practices. Testing a piece of experimental software for in-house use, for example, is quite different from testing a software product to be released for sale to the general public, although the practice of testing a program may be identical in all other respects.

Similar, less technical, examples abound at all levels of human interaction. One says "Hello" to a friend differently from the way one does to a stranger in an elevator; one hugs one's sibling differently from the way one hugs one's spouse; one writes a letter to one's aunt differently from the way one writes to a customer service department, etc.

This information, therefore, is quite important to the representation of the expert's actions. Interestingly, it is often less easily accessible, for what one is doing by doing *this* is often simply part of the "ground" on which the "figure" of one's current action is taking place. As we shall see, this is another area in which the knowledge engineer's interviewing skill is particularly important.

The Community

At this point we have discussed most of the types of information present in the interview with the expert programmer. However, there are other aspects of describing a person's actions that we must address. Social practices have a place in the larger configuration of a community (Putman, 1981). Certain aspects of the description of the community in which the informant's actions have a place are important in producing a system to engage in those actions. The most important of these are the statuses involved in the practices, and how intrinsic the practices are to a person in each Status.

As discussed by Putman, one's status in a community is a codification (a representation) of one's eligibilities for various practices. For example, "programmer" is the name of a certain status (or "place") in what one might call the "programming community", and "debugging a program" is what members of that community call the practice our expert informant was discussing.

To a person in some status within some community, certain practices are engaged in witb no further end in view. To put it another way, they are done for their own sake, rather than as part of some larger practice. Such practices are called *intrinsic*. To a programmer, writing a program (i.e., designing it, writing it, and debugging it) is intrinsic.

Other practices are *instrumental*, that is, done not for their own sake but because they are a part of a larger practice. For example, running the program with a bug several times is not something a programmer does for its own sake; it is a stage in debugging the program.

The expert system to be produced is to function in certain ways as an expert in the area of interest. This means that it will have a certain status (place) in the community in which the practices in which it engages have a place. That place is an aspect of the information the knowledge engineer must gather.

OTHER APPROACHES

The approach that has been presented differs in two important ways from more traditional ways of organizing expert knowledge.

The first is to focus on the action as the central logical element. The social practice description, as described in the foregoing, is the vehicle for representing the social practices the expert, or the expert system, engages in. The process structure of the practice is given by the stage-option structure, as controlled by the constraints. The roles that objects play in the practice is represented by the element-individual-eligibility structure. Knowledge is "defined" by having the place it does in the social practices: concepts, facts, and perspectives necessary in order to carry out the practice.

The second difference is that while it may be necessary in some cases to specify how some item of knowledge is found, this is not the central focus. The focus is rather on where in the practice being described the knowledge makes a difference. The traditional approach virtually always assumes that knowledge has a certain structure, and is inferred or deduced by various mechanisms. (Good examples of this philosophy are Brachman, 1979 and Weilinga & Breuker, 1985.)

Frames and Scripts

In recent years certain researchers have noticed that is often useful to assemble these items of information into "chunks" (as they refer to them), called "frames". A frame is a collection of items and properties that together describe an object or event. When the frame is oriented to describing an action, it is a "script" (Rich, 1983; Winston, 1984).

There is clearly a good deal of similarity between a frame and a social practice description. This is probably due to the fact that both approaches address the problem of describing the actions of a person, and there are severe logical constraints on what must be included if one is to have an adequate account of a person's action (Ossorio, 1970/1981).

Social practice descriptions differ from frames primarily in content, rather than formal structure. The concept of a frame (as it is currently understood in AI) is cruder than that of the social practice description. It is designed to represent answers to the question, "What plays a role in this thing (action, object, etc.)?" Any process, object, or state of affairs that plays a role may be included. For example, the flame on the candle on the birthday cake, the ribbon on the birthday gift package, and cutting the cake are all typical elements in a frame describing a birthday party (Rich, 1983). One specifies a frame simply by specifying its name and its parts, known as "slots".

Social practice descriptions are designed to represent everything one can say about a practice, at this level of detail. The various parts of the description have the structure, and relations, given above. Another difference lies in the use of the concepts: merely having the formal structure of a social practice does not mean that the description describes a practice; the knowledge engineer must determine the "recognizable patterns of action" that comprise the practices of the community under discussion (Ossorio, 1970/1981). A birthday party, for example, is not a single social practice, but several. The flame on the candle is an individual in a certain practice (the one with the name "blowing out the candles"), which may be a stage in a larger practice. This of course does not mean that frames, and their use, could not be refined to the point where they were virtually the same as social practice descriptions.

There is a further difference which needs elaboration, again having to do with use of the two concepts. The traditional use of frames has been to organize knowledge, not to represent the actions to be done (by the expert or the expert system) and determine what to do next. As noted earlier, actions are treated as implications of knowledge. Examples are R1/XCON, a system used by the Digital Equipment Corporation to configure computer systems (Rich, 1983) and MYCIN (Harmon & King, 1984), Weilinga and Breuker (1985), and Boose (1985).

This difference is evidenced in two ways: how the knowledge engineer gathers the expert knowledge, and the form of the representations produced. The knowledge engineer using the approach of this paper begins with the actions (the practices) the person engages in. Rather than making assumptions about the knowledge necessary to engage in the practice, or about the structure of such knowledge, or about mechanisms for using such knowledge, or indeed whether it makes sense even to talk in these terms at all, the knowledge engineer begins with the most conservative question: "What can we say about it?"

This question, and elaborations, are the topic of the next section.

THE PRACTICE OF KNOWLEDGE ENGINEERING

This section discusses how one goes about gathering information from an expert. It relies heavily on the analysis of action given in the foregoing section. It endeavors to give some organized, detailed, and useful information, of which there is comparatively little in the published literature. However, a disclaimer is in order: it does not attempt to present, nor even to indicate, a procedure (in the sense of series of steps that one do with the ordinary expectation of success) for knowledge engineering.

The knowledge engineer is a person attempting to gain information about another person's actions, including not merely the overt steps (the versions), but the necessary distinctions the expert acts on, the perspective(s) he adopts, all of the constraints covering all of the combinations of stages and all of the eligibilities, the place this practice has in larger practices, the values the expert is expressing in the practices (the choices he makes), and the language the expert uses. In short, the knowledge engineer is gaining both broad and deep insight into an area of a person's life. This is exactly the sort of endeavor in which one would expect tips, reminders, rules of thumb, and a good deal of skill to be involved, rather than a procedure with an assured outcome.

Actor, Observer, and Critic

As the above section discusses, there are several aspects, or types, of information one must provide to describe the behavior of an expert. Rarely can one simply ask a person directly for the information, for two reasons. First, they usually cannot tell you. It is virtually always that case that a person's ability to act far outstrips his ability to describe his actions. This often seems paradoxical to expert system builders, but is a straightforward reflection of the reality of human behavior: describing a practice is itself a second practice, and there is no reason to expect the concepts, facts, perspectives, stages, elements, eligibilities, or significance of the second to be the same as those of the first.

This difficulty can surface in a slightly more subtle form: It is not likely to be effective to ask an expert for facts and beliefs he has about a subject matter area, although this is a very common way of proceeding (Hardy, 1986).

One might paraphrase this by noting that as soon as I ask you what you are thinking, what you are thinking changes.

Since one cannot simply ask, what can one do?

A brief answer is that one relies on the distinction between Actor, Observer, and Critic (Ossorio, 1970/1981). It is not necessary for a person to observe and describe his own behavior. Rather more common is to have a different person giving descriptions, and this is the paradigm case for knowledge engineering: The expert acts, in his area of expertise, and the knowledge engineer observes and describes the expert's actions.

As he interviews the expert, the knowledge engineer uses the distinctions elaborated in the previous section to recognize where further information is needed and, more generally, to formulate a description, or representation, of the practices as the expert engages in them. The knowledge engineer then takes all the data and produces a description covering all of the information he has, in the technical format required. This description will include the practices (specified by name), and descriptions of those practices.

Where to Begin

As with other social practices, the practice (or practices) of knowledge engineering requires a certain perspective, namely that of observer/describer of human action. This may seem painfully obvious, but in knowledge engineering as it is usually done this perspective is often confused with others. The two most common are the theoretician giving theoretical accounts involving hypothesized mechanisms, and the computer scientist giving accounts in terms of symbolic information processing. (See, for example, Firschein, 1984, Brachman, 1979, or virtually any issue of the *Artificial Intelligence Journal*).

As an observer/describer, the observable performances are to give descriptions of behavior, in this case the behavior of an expert. The facts are the observable episodes of the expert's behavior, his performances. The concepts used are those of the social practice and social practice descriptions, as articulated in above. The primary criterion by which a description is judged is whether it is an informative, useful description of the behavior; abstraction, theorizing, interpretation, and mechanistic modelling have no place in the knowledge engineer's action.

As in most interviewing situations, one begins with a simple, broad question, sucb as "Tell me about diagnosing thyroid problems". If tbe expert can respond to the question, the knowledge engineer has begun to get his data. If not, he may try a more specific question, based on his understanding of the expert's field. Frequently, a very useful technique to get started is to "mock one up", or ask the expert to pretend that a typical case has come up and then show the knowledge engineer what he does.

During and after the interview, the knowledge engineer looks for those "recognizable patterns of action" that are the practices the expert engages in. These may be observable ("Send in a report", "have the diagnosis verified", etc.) or not ("Decide how many segments to divide this program patch into", "Assess whether my subordinate is motivated by teamwork", etc.).

A technique that is frequently very useful is to begin with a request for an overall description of what the expert does. The answer to this question will typically give, by name, the highest-level social practices the expert engages in. Then, with more detailed probing, the practices that make up the stages of the higher-level practices are named. The knowledge engineer then has the task of recognizing whether there is a gap between the two levels of description he has so far and "filling in" if so.

To describe the identified practices, one describes the stages, and the versions, involved in carrying out them out. The expert, in engaging in a behavior or in talking about it, will refer, or on occasion explicitly mention, knowledge he uses: facts and data, concepts, and perspectives. Often this will be in the context in which the knowledge is used.

Thus, the expert does any of the variety of things a person does in which they use their expertise. Typically, one of these is talking about what they do, and this is most often the easiest place to start. The knowledge engineer questions, probes, requests elaboration, and prompts the expert.

A partial list of the skills that make a difference in being able to carry out this practice are knowing when and how to probe, how to prompt, how to get more detail, and how to feed hack one's understanding so that the expert can meaningfully verify it. These are in addition to recognizing when a person is referring to, or doing, a separate social practice, and recognizing the various aspects of actions: Knowledge, skills, stage-option structure, and the element-individual-eligibility structure. Further, there are interpersonal skills such as being able to recognize whether the person is comfortable in the interview and being able to recognize whether the person has more to say but would like to continue at another time.

It is not clear, to this author at least, how much value some knowledge of the expert's domain is in the knowledge engineering process. It seems clear that some knowledge is valuable, for the expert may not refer at all to some area, fail to cover various cases, and so on, and in such cases the knowledge engineer is in better position to recognize the errors. What is not clear is the degree to which this knowledge can be gained in the course of interviewing the expert, or experts, and to what extent a total lack of domain knowledge hinders the knowledge engineer.

Let us consider an example:

> An expert reports, "If this is a Type-3 failure report, then I route it to Jane, but Type-4's go to Dave in the next building".
>
> The concept of Type-3 vs. Type-4 report is used; it appears in a constraint on a stage or options of a stage. (It is unclear from this fragment which is the case.) However, it is clear that some different action is taken in the two cases, so Type-3 vs. Type-4 will appear in some constraint. One does not know, at this point, Jane and Dave's roles in the practice—the elements for which they are eligible in the practice. Thus, the knowledge engineer notes several questions: (a) What are Jane's and Dave's jobs, (b) What is the informant doing by sending the report to these people, (c) Under what conditions is each of the actions taken, (d) What will Jane and Dave do with the report (that is, what practices will they engage in), and (e) What larger practice is this a stage of.
>
> The second question needs elaboration. "Send the report" is a performance description, that is, one that gives no information as to the practice being engaged in. That practice, in turn, may be a stage of a larger practice (as noted earlier), but so far the knowledge engineer does not even know the practice. For example, it might be that Jane is the programmer responsible for the code addressed in the failure report. In such a case, "Sending the report to Jane" is the practice of "Notifying the responsible programmer of a problem". Or, "Sending the report to Dave" might be a case of the practice of "Filing an erroneous report with the Program Clerk".

Thus, having the concept of social practice, and social practice description, the knowledge engineer has a great deal of information about what to look for: the practices the expert engages in, and how the structure of those practices. Since the aspects of the social practice description have the relationships discussed earlier, recognition of one part (e.g., an item of knowledge) leads to questions about how that item is used (the constraint), and in turn the stage-options, or eligibilities, in which that constraint has a place. The social practice description can be seen as the template for the wholes into which the parts that the expert supplies fit. The knowledge engineer's task is to represent those wholes, their parts, and all of their relationships.

An additional skill the knowledge engineer must have is to be able to recognize when he has a complete and coherent account from the expert. Often he must additionally be able to recognize when he is getting information he will be able to use in producing a useful representation, but before he has that representation in hand.

It is not uncommon to reach a point at which the expert has difficulty in further verbal description of his actions. When this occurs the knowledge engineer may ask the expert to "act it out", pretending he is actually engaged in the practice, while the knowledge engineer continues to observe and describe the actions. The knowledge engineer may play the role of another person (when one is involved in the practice of interest), asking questions as a person *in that role* does. The expert's actions then form the "raw data" for further observation and description.

The process just outlined is in many cases similar to the usual ways of proceeding in gathering information from experts. It differs in certain significant ways. First, it is neither interpretive nor self-report in nature; it could reasonably described as "division of labor": the expert acts, and the knowledge engineer describes. However, it is crucial that the knowledge engineer have available descriptive resources adequate to the task of describing complex human action. This is the role of the social practice description; it provides a technically usable framework for representing all the facts about any human action that does not force the describer to abstract, invent, interpret, or otherwise change the content.

Second, one makes use of self-report data as, and when, it arises, identifying the actions, concepts, and skills mentioned in the report. The practices will be additional practices not yet covered, stages (or sub-stages) of other practices, or larger practices in which already known ones have a place. The concepts and skills will have a place within these practices, as discussed earlier, or may refer to additional practices.

Experience with Social Practice Descriptions

The approach presented in this paper has been used to produce several actual expert systems. The projects ranged from highly technical practices in the construction and change of very large computer programs (Jeffrey & Putman, 1983) to the very "soft" endeavor of consulting with a manager on the people-oriented practices of management, such as improving someone's job satisfaction or getting another manager to cooperate on a project.

Proceeding is this way has been much more efficient that the more usual approaches in the area. One of the very significant costs of building an expert system is the expert's time. It is generally accepted that to capture someone's expertise in a fairly large area will require full-time involvement by the expert for several months. Experience to date is that this approach allows us to cover a comparable breadth and depth of expertise in approximately a few weeks of the expert's time.

The direct involvement by the expert is in several phases. Initial interviews typically take at most two to three hours; one to two hours is by far the typical case. After initial descriptions are produced, it may take another two to three hours with the expert to check them out for accuracy, completeness, tone, direction, and overall consistency. Further elaboration may take several more hours, in blocks of one to two hours. Close monitoring of the language of the descriptions, with the expert, may again take several more hours.

The data so far of course do not constitute a controlled experiment. Gathering more data on the time and costs of various approaches to knowledge engineering, including this one, is an area for further work.

A further advantage of this approach is that the involvement by the expert, in addition to being approximately one tenth to one fifth of the total time needed for other approaches, is not in one single full-time block. (In actual practice this is very important; experts rarely have a two-week block of time easily available.)

Practical Heuristics

Ordinary English includes a variety of ways of expressing the distinctions one needs as a knowledge engineer. Presented here are a few of the rules of thumb and tips that have been found to be useful. The primary value of this discussion, I believe, is to alert the reader to the careful analysis of a person's language, another very important skill for a knowledge engineer. A careful, systematic study of this area would seem to be an interesting topic for further research.

Knowledge. Part of what the knowledge engineer relies on to recognize the concepts—the facts, distinctions, and perspectives—that play a part in the expert's actions is the expert's language.

It is common for people to use language such as "having an idea of . . .", "an image", "getting the feel of . . ." and other phrases referring to mental, physical, or emotional aspects of doing some practice. We have found that very frequently these locutions refer to an item of knowledge.

As noted earlier, often information on necessary perspectives is the subtlest, or most difficult to acquire, of the knowledge needed. Sometimes this is not the case; the informant will refer overtly to it. For example, an expert computer system designer may report, "You have to understand how the user is going to view the system." More commonly, the informant will use phrases like, "frame of mind" or "outlook".

Perhaps most common in this area are phrases that appear to be general statements or policies. One will often encounter phrasing such as, "the basic thing you are after here is . . .", or "what you are looking for here is . . .".

A useful rule of thumb is that in addition to such obvious linguistic clues as "point of view" or "outlook", consider whether the informant is referring to a perspective whenever such terms such as "overall" or "basically" are used.

This is, however, only a rule of thumb, because such terminology is also frequently used to refer to the stages of a practice, or part of it. It is also found when the informant is emphasizing the outcome of a practice, rather than its stage structure—i.e, giving an achievement description (Ossorio, 1970/1981).

Significance. Frequently one must specifically inquire for the information on the significance of a practice. It is rare to be able to ask directly, "What larger thing are you engaged in here?" Usually, phrasing the question as, "When do you do this?" or "Under what circumstances do you do this?" is more effective.

It is not uncommon, however, for an informant to give what would be answers to this question in the course of talking about it. One often hears, "Well, when I come into the picture is when . . .", or similar language. Again, there is considerable ambiguity here, because the informant may be referring simply to a stage of the practice.

Skills. There seem to be two ways in which people typically talk about skills involved in what they do. First, as they discuss what goes on as they are engaged in the practice, typically a skill will sound like a stage (that is, a separate step), but one which happens all the time and at the same time as other stages.

Second, when the informant is asked how he does one of these things, or how it looks when he is doing it, the responses tend to be, "I just do it, that's all", "I don't know what I do, I just do it", or, perhaps most expressively, a blank look.

This is an area in which it is easy to distort the informant's information if one is not sensitive to the informant's own stopping point—the point at which, for that person, there is in fact no "how". Further, this point will vary from person to person. As a person's expertise in an area increases, the number of cases in which the person simply knows how increases.

When one encounters a skill, it is not necessarily the case that there is nothing further to say. There may not be any stages involved, but there may well be knowledge. Often the expert can say a good deal in response to a question such as, "When you do this, what is important here? What do you pay attention to?" Sometimes a less pointed question is helpful in these cases: "OK, let's imagine you have one of these cases. What would you say about it?"

OTHER FORMS OF INTERVIEWING

Knowledge engineering is certainly not the only human endeavor in which one interviews another person to gain an in-depth understanding of what he does, how he does it, and how he looks at things. A brief examination of two such areas may shed light on the perspective needed for this one.

Perhaps the most direct similarity is to journalism. A journalist must gain information from, and understanding of, the person he is interviewing that is very similar to that needed by the knowledge engineer. We believe that the reason for this is that, once again, the logic of describing a person's actions is fundamental. The knowledge engineer and the journalist both have the job of gaining this information and communicating it to others. (In fact, in the past we have preferred the term "technical journalism", due to A. O. Putman, for the enterprise we are engaged in, because it seems considerably broader and more descriptive.)

The second area with a notable similarity to knowledge engineering is one familiar to those with a psychological background: clinical interviewing. Here again is an area in which the outcome is an in-depth understanding of a part of a person's life. Many of the interviewing techniques are quite similar. In fact, we have found that background in clinical psychology and interviewing is extremely helpful in this work, due in part to the experience one gains in unobtrusively finding out how a person does things and sees things. The focus of a clinical interview is of course different, but many of the concepts, perspectives, steps one takes, and skills needed are the same.

SUMMARY

The practicing knowledge engineer has the job of producing complete, precise, technically useful representations of human behavior. As a complete, precise, systematic formulation of the concepts of person, behavior, language, and the real world, one would expect that Descriptive Psychology would have a good deal to offer to the knowledge engineer, and the facts have borne out this expectation. In the last several years considerable experience has been gained in using the formulations of Descriptive Psychology to do knowledge engineering in a variety of areas. This paper has presented some of these formulations, and the concepts, approaches, and practices based on them.

REFERENCES

Boose, J. H. (1985). Personal Construct Theory and the Transfer of Human Expertise. In T. O'Shea (Ed.), *Advances in Artificial Intelligence* New York, NY: North Holland.

Brachman, R. J. (1979). On the Epistemological Status of Semantic Networks. In N. V. Findler (Ed.), *Associative Networks*. New York, NY: Academic Press.

Buchanan, B. G., and Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley.

Duda, R. O., and Reboh, R. (1984). AI and Decision Making: the PROSPECTOR Experience. In W. Reitman (Ed.), *Artificial Intelligence Applications for Business* Norwood, NJ: Ablex Publishing.

Ferrand, P. (1985). An Explanatory Medical Aid System. In T. O'Shea (Ed.), *Advances in Artificial Intelligence* New York, NY: North Holland.

Firschein, O. (Ed.) (1984), *Artificial Intelligence*. Reston, VA: AFIPS Press.

Harmon, P., and King, D. (1985). *Artificial Intelligence in Business*. New York, NY: John Wiley and Sons.

Hardy, S. H. (1986). Lecture given at Northern Illinois University, October 14, 1986.

Jeffrey, H. J. (1981). A New Paradigm for Artificial Intelligence. In K. E. Davis (Ed.), *Advances in Descriptive Psychology* (Vol. 1, pp. 177-194) Greenwich, CT: JAI Press.

Jeffrey, H. J., and Putman, A. O. (1983). The MENTOR Project: Replicating the Functioning of an Organization. In K. E. Davis and R. Bergner (Eds.), *Advances in Descriptive Psychology* (Vol. 3, pp. 243-270) Greenwich, CT: JAI Press.

Kraft, A. (1984). XCON: An Expert Configuration System at Digital Equipment Corporation. In P. H. Winston and K. A. Prendergast (Eds.), *The Commercial Uses of Artificial Intelligence*. Cambridge, MA: The MIT Press, 1984.

Ossorio, P. G. (1971/1978). *"What actually happens"*. Columbia: University of South Carolina Press, 1978. (Originally published in an earlier version in 1971 as LRI Report No. 10a, Whittier, CA and Boulder: Linguistic Research Institute. Later listed as LRI Report No. 20.)

Ossorio, P. G. (1970/1981). Outline of Descriptive Psychology for Personality Theory and Clinical Applications. In K. E. Davis (Ed.), *Advances in Descriptive Psychology* (Vol. 1, pp. 57-81) Greenwich, CT: JAI Press, 1981. (Originally published in 1970 as LRI Report No. 4d. Whittier, CA and Boulder: Linguistic Research Institute.)

Ossorio, P. G. (1981/1983). A Multicultural Psychology. In K. E. Davis and R. Bergner (Eds.), *Advances in Descriptive Psychology* (Vol. 3, pp. 13-44) Greenwich, CT: JAI Press, 1983. (Originally published in 1981 as LRI Report No. 29. Boulder: Linguistic Research Institute.)

Putman, A. O. (1981). Communities. In K. E. Davis (Ed.), *Advances in Descriptive Psychology* (Vol. 1, pp. 195-209) Greenwich, CT: JAI Press.

Putman, A. O. (1985). Managing Resistance. In C. R. Bell & L. Nadler (Eds.), *Clients and Consultants* (2nd. ed.). Houston: Gulf Publishing Co.

Rich, E. (1983). *Artificial Intelligence*. New York, NY: McGraw Hill, 1983.

Schank, R. C., and Riesbeck, C. K. (Eds.) (1981). *Inside Computer Understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Wielinga, B. J., and Breuker, J. A. (1985). Interpretation of Verbal Data for Knowledge Acquisition. In T. O'Shea (Ed.), *Advances in Artificial Intelligence*. New York, NY: North Holland.

Winston, P. H. (1984). *Artificial Intelligence*. Reading, MA: Addison Wesley.