The Knowledge Engineering Review, Vol. 4: 3, 1989, 235–247

Explanation and dialogue

NIGEL GILBERT

Social and Computer Sciences Group, Department of Sociology, University of Surrey, Guildford GU2 5XH

Abstract

Recent approaches to providing advisory knowledge-based systems with explanation capabilities are reviewed. The importance of explaining a system's behaviour and conclusions was recognized early in the development of expert systems. Initial approaches were based on the presentation of an edited proof trace to the user, but while helpful for debugging knowledge bases, these explanations are of limited value to most users. Current work aims to expand the kinds of explanation which can be offered and to embed explanations into a dialogue so that the topic of the explanation can be negotiated between the user and the system. This raises issues of mutual knowledge and dialogue control which are discussed in the review.

1 Introduction

One reason often heard for favouring knowledge based systems over more conventional programs is that they can explain their decision making. Knowledge-based systems, by virtue of their explicit representation of knowledge and their capacity to provide a trace of deductions, are alleged to be capable of providing the explanations that users need to be able to make their own judgements about the program's recommendations. However, although some capacity for explanation was included in the earliest expert systems, experience with these merely pointed to the difficulty of automatically generating explanations which were comprehensible and useful. What are good explanations, how they might be constructed and, especially, their role in a dialogue between user and system are continuing matters of investigation and debate. This review outlines the main issues and current research directions, focusing primarily on generating explanations from advisory knowledge-based systems (systems which provide tailored advice to the user on, for example, medical diagnosis, financial planning and legal requirements).

Initially, the task of explanation for advisory systems was taken to consist of displaying a record of the deductions used to obtain a conclusion, a "proof trace", and there continues to be considerable work on manipulating such traces to make them more comprehensible to users. The paper begins by briefly reviewing this approach and then argues that explanation encompasses much more than can be derived from a proof trace. One of the problems of explaining is working out just what the user wants to know. This can involve classifying the types of explanation available, making assumptions about the user's current state of knowledge and engaging in a dialogue with the user about what is needed and when sufficient explanation has been provided. These issues are reviewed in subsequent sections of the paper, where it is suggested that an explanation should not be thought of as an item presented to the user by the system, full-formed, but rather a jointly negotiated, emergent feature of a dialogue between the system and user.

2 Explanation as proof trace

Expert systems from MYCIN onwards (Shortliffe, 1976; Scott et al., 1977) have provided users with the opportunity to ask "How?" and "Why?" and have interpreted the questions to mean "explain how you reached this conclusion" or "explain why you are asking me that". Answers could be generated easily as a by-product of the backward chaining inference mechanisms typically used by these systems: the current stack of unresolved goals gave an "explanation" of why the current question was being asked and a record (the "proof trace") collected during inference yielded an explanation of how an answer had been deduced. However, the early proof trace explanations suffered from a number of problems. First, because the explanations were generated as a by-product of the inference mechanisms, their structure reflected the structuring of the knowledge in the system's knowledge base. Often, this did not result in an explanation which was easy for users to understand. Second, the explanations tended to be verbose. They included references to internal procedures (for example, arithmetic operations) and they were repetitious: if a deduction was made more than once in the inference chain, both were reported (Swartout, 1977). Third, the granularity of the explanation was often wrong: when the user wanted a broad overview, the system could offer only a minutely detailed account. Fourth, the explanations were not in any way designed to match the user's expectations or prior knowledge.

Weiner's BLAH system (Weiner, 1980) was one of the first to attempt to tackle these problems in an integrated way. BLAH maintained a "user's view" and a system's view of its knowledge base. These segmented the knowledge into those assertions and deductions which the user was assumed to know already and which could therefore be deleted from an explanation, and those which were known only to the system. BLAH's explanation was based on a proof trace which was subjected to a series of transformations. Once the explanation had been "pruned" of items already known to the user, its structure was transformed using a set of rules which, for example, broke a complex explanation into a number of sub-explanations together with linking phrases. Finally, the explanation was passed through a text generator to produce English sentences. As well as asking "Why?" questions about deductions, users could also ask for explanations of why one recommendation was preferred by the system over another.

BLAH paved the way for subsequent work on explanation in a number of respects. It hinted that there was more to explanation than just proof traces. It pointed to the importance of tailoring the explanation to suit the user's needs and knowledge. And the transformational rules were devised from a study of "natural" explanations, that is, of people explaining to each other. This emphasis on learning from the human capacity to explain has been a strong undercurrent in all subsequent work.

3 Explanation-based generalization

Proof trace explanations have recently also become important as the starting point for explanation based generalization (EBG) and its variants, explanation based learning and case based reasoning (Harman, 1986; Pratt, 1987; DeJong, 1988). For EBG, an explanation is generated by backwards chaining through many domain rules to prove a given goal (the "example") in exactly the same way as it would be if the explanation had been requested by a user. Then this trace, consisting of a tree of instantiated rules, is "generalized" by changing any attributes of the goal which are not involved in the proof into variables. The generalization explains other goals which are the same as the given goal in all respects relevant to the proof, but which differ in irrelevant ones (Mitchell et al., 1986). Finally, the generalized explanation is added to the system's rule base (i.e. it is "learnt") so that next time a goal which resembles the original one is encountered, the newly learned rule can be applied without needing to repeat the possibly lengthy chain of inference which was used to solve the original goal.

EGB has some features worth noticing for their relevance to computer–human explanation. In EBG, an explanation is only usefully learnt after it has been generalized. The generalization algorithm depends on access to the domain rules which generated the trace, in order to see what should be generalized (that is, how variables should be substituted for the constants in the trace). A person receiving a MYCIN-type explanation consisting of an instantiated proof trace has to perform the same kind of generalization. This will only be possible if he or she already knows a lot about the rules which were used to generate it. A proof trace explanation may therefore be helpful to a knowledge engineer debugging the knowledge base, but of limited value to the typical user. The latter would need both the trace and the rules which were involved in creating it.

EBG has trouble with explanations of failure, for example, in answering questions such as "Why is X not true?" or "What is it that does not allow X to be deduced?" (Rousset and Safar, 1987). Because the proof trace only records those rules which fired, rules which were tried but which failed to match are not noted. In fact, the problems with answering such questions are more fundamental than merely the absence of information in the proof trace. Some criterion of relevance has to be brought to bear to answer negative questions. Even under the closed world assumption (that all the necessary knowledge is known), there will be an indefinite number of reasons for failure, although all but a few would appear irrelevant or contrived if presented to the user. The difficulty lies in giving the system sufficient knowledge to allow it to work out what counts as a relevant explanation. The same problem does not arise in explaining success, because here there is usually only one explanation, the trace of the deductions which led to the answer.

4 Answer types

Although the proof trace has been widely regarded as the archetypal form of explanation, studies of transcripts of human explanation, for example from engineers doing fault diagnosis and experts answering listeners' questions on radio phone-in programmes (Pollack et al., 1982; Kidd, 1985; Crossfield, 1986), have drawn attention to the fact that human explanations come in many other forms and provide a wide range of different kinds of information. Some way of organizing and categorizing these many types of explanation is needed. The first proposals focused on the wording of the question (so, as with MYCIN, one had "Why?" explanations and "How?" explanations (Graesser and Murachver, 1985)) or on the underlying meaning of the question (for example, is it a verification, enablement, disjunctive, judgemental, etc., question (Lehnert, 1977; Hughes, 1986)). In an earlier paper (Gilbert, 1987a), I suggested that it is unhelpful to classify on the basis of the question, which is often ambiguous, and that it is better to use a classification of "answers", that is, of the explanations themselves, based on the kind of knowledge from which each can be derived.

Table 1 is reproduced from Gilbert (1987a) with some modifications arising from experience using the classification. The knowledge base is divided conceptually into three levels (the rows of the table) and four kinds (the columns). The case level includes knowledge specific to the particular case under consideration; the domain level includes, as its name suggests, domain knowledge, and the theory level concerns knowledge about the domain level, that is, what is often called meta-level knowledge (Clancey, 1983). The four kinds of knowledge which are distinguished are: taxonomic, knowledge about entities and their relationship; formal, analytic knowledge about relationships between entities' attributes; contingent, synthetic knowledge about causal and temporal relationships; and control, knowledge about the system itself and its inferencing.

The cross-classification identifies twelve types of knowledge which can be manipulated in various ways (thus giving rise to more than one answer type in some cells) to provide a full range of explanations. The scheme has been useful in a number of studies (Gilbert, 1987b; Nicolosi et al.,

Table 1 Answer types (adapted from Gilbert, 1987a)

	Taxonomic	*Formal*	*Contingent*	*Control*
Theory	existence	warrant definition	process method	strategy capability
Domain	generalization exemplification identification	relevance	implication	
Case	instantiation	classification prescription sensitivity	procedure antecedent	justification rationale

1988) and has been implemented in the "Advice System", an expert system for advising people about social security benefits (Gilbert et al., 1989; Robinson, 1988).

One implication of this type of classification exercise is that knowledge based systems which aim to provide explanations will need a rich knowledge base. Typical advisory expert systems are able to produce classification and prescription answers, while the How? and Why? questions of MYCIN tap the justification and rationale answer types of Table 1. The remaining answer types are not usually supported, because the knowledge needed to generate them is either missing from the system or is "compiled away" and inaccessible to the inference mechanism and answer generator (Swartout, 1983).

5 Planning what to say

In addition to a rich body of knowledge about the domain, the system also needs knowledge about how to create clear explanations. This means that it needs to know about effective explanation structures, in what order to present the components of an explanation and how to convert an explanation couched in some internal representation into comprehensible natural language. The most common approach to these issues is to use "rhetorical schemas" (McKeown, 1985) or "grammars" (Goguen et al., 1983) to define what can be said and how it should be organized, and a hierarchical planner to generate a semantic representation of the utterance. This representation is then passed through a text generation component to produce natural language output.

The rhetorical schemas and grammars have generally been derived from transcripts of explanations constructed by humans. For example, Cawsey (1989; see also Maybury, 1988) has examined explanations offered by instructors and proposes Figure 1 as a typical way of explaining a component in an electronic circuit diagram.

Explain → Describe, Explain behaviour, Summarize
Explain → Type of circuit, Summarize
Describe → Type of circuit, Function, Describe Components
Describe → Analogy
Describe component → Function, I/O behaviour
Function → I/O dependencies
Type of circuit → Circuit type, Functional type
Explain behaviour → Explain dependencies, Explain I/O
Summarize → Function, I/O behaviour

Figure 1 Explanation grammar (from Cawsey, 1989)

In her system, an explanation of a component can consist of a "description" of the component, followed by an explanation of the component's behaviour, and a summary. A "description" can consist of a description of the generic type of a circuit, a statement of its function and a description of its sub-components, or can be a description in terms of an analogy. Because there are many alternatives in her grammar (for instance, one can either describe by describing the sub-components or by using an analogy), an explanation can be tailored to the user and to the conversational context.

In practice, the schemas are often converted into "operators" for a text planner. Each operator indicates the circumstances in which it may be used (the "preconditions"), the effects its use produces (the "postconditions") and the operations which must be carried out in order to apply it (the "actions"). For example, it is not sensible to describe an electronic component by analogy to another component unless there is evidence that the explainee understands the other component; such understanding would be a precondition to using the "describe with analogy" operator. In order to describe with analogy, it is necessary to name the component concerned and to point out the similarities (and possibly the differences), these being the "actions" for the operator. The effect of applying the operator is that the component has been described, thus perhaps making it unnecessary to redescribe it later in the dialogue.

Given a goal, such as "explain this component", a planner should be able to find a plan consisting of a network of operator nodes in which all the operator preconditions are respected. The plan defines a sequence of basic actions (e.g. "name the component") which are acceptable to a text generator and from which the generator will form a well structured and appropriate explanatory utterance. The goal which the planner tries to satisfy by constructing a plan will be determined by the explainer's view of the topic and the kind of explanation required. By including several explanation schemas and accommodating several different kinds of goal, it is possible to build a system which will offer quite different explanatory texts depending on the requirements of the situation.

Although the text generator only needs the lowest level actions, corresponding to the terminals of the schema grammar, in order to produce explanatory text, there are advantages in producing and storing the whole plan network (Moore and Swartout, 1988). This network shows how each operator was expanded into actions, the dependencies between nodes enforced by operator preconditions, and the effects of each operator. The text generator can search the network to locate the intended "focus" and use this information as a simple way of controlling the substitution of anaphora for referring expressions. For example, if the objective is to explain an AND gate, the text can refer to the gate as "it" until the explanation becomes involved with a sub-goal to explain a transistor forming a component of the gate, whereupon the transistor becomes the focus and the gate will need to be referred to explicitly.

The plan network is also valuable if the user either interrupts an explanation, or seeks clarification afterwards. On resuming after interruptions, the system can reconstruct what it was doing and why by examining the network. In responding to requests for clarification, it can use the network to identify the topic that needs clarification and to recall how the explanation was previously provided, so that a new, more extended or alternative version can be planned.

6 Occasions for explanation

Providing the machinery for rich, carefully composed explanations is of little value unless users have some reason and the opportunity to request them. Studies of users' desires to obtain explanations have come to inconsistent conclusions: those which involve the analysis of human–human dialogues tend to find explanation permeates the whole interaction (Gilbert, 1987b; O'Malley, 1987a), while in human–computer protocols, explanations seem to be few and far between (even when, as in experiments carried out by Rogers (1988b), the "computer" was in fact simulated by a human in a "Wizard of Oz" type of experiment).

There could be several reasons for these differences. Firstly, computers do not have a reputation for providing helpful explanations and so perhaps users fail to ask for what they don't think they will get. Secondly, human–computer interaction is inevitably a strange and unusual social context, which may inhibit the dialogue, directing it towards imperatives rather than explanations (Diaper, 1986; Guidon, 1988). Third, most knowledge based systems are not yet capable of providing explanations which users would find illuminating and helpful, and so users don't ask.

Even if we can overcome the last problem and provide useful explanations, it will still be necessary to ensure that users have an appropriate image of the system and its capabilities. What they then ask for will depend on the social and technical context in which the system is embedded and on the users' perceptions of their need for explanation.

Why do people think that they need an explanation? Except in certain situations (for example, examinations, scientific proof and legal proceedings), explanations seem to be desired only when there has been some kind of "expectation failure" (that is, when a new data item gives rise to surprise or paradox). O'Malley (1987b) (see also Ortony and Partridge, 1987) identifies three sources of expectation failure: an observation conflicting with what was actively predicted; one conflicting with what had been passively assumed; and one conflicting with a retrospective expectation. In the former two types, the surprise arises because what had previously been thought to be the case turns out not to be so. In the latter, no prior expectations or assumptions had been made, but when the new observation comes about, it is found to be "abnormal", that is, no plausible set of hypotheses can be found by abduction to explain it.

It is the latter kind of expectation failure which is likely to prompt the user of a knowledge-based system to ask for explanations. Because users are generally interacting with the system to seek advice or discover something new, it is improbable that they will come with either active or passive predictions of what the system will advise. Hence, most explanations are triggered when users try, retrospectively, to account for or "understand" the system's output and find themselves either unable to do so, or able only using rules and concepts which conflict with their own beliefs. An important part of creating the need for a system explanation is therefore the user's attempt to generate an explanation for him or herself. Only when no explanation can be found, or its validity is very doubtful, will the user seek an explanation from the system or from another source.

If the extent of the user's knowledge were sufficiently precisely known, the system itself could in principle work out whether the user would be capable of explaining a particular answer, and, if not, it could accompany the output with an explanation, anticipating the user's request. This would be seen by an observer as the system producing a "spontaneous explanation" (Draper, 1987), a notion which will be considered further below. An ability to anticipate the need for explanations is clearly desirable, but it depends on being able to ascertain the state of the user's knowledge. This is one of the main tasks for which user models are advocated.

7 User models in explanation

That the system should have a "model" of the user has seemed a self-evidently good idea ever since it was first proposed. Such a model, in the context of providing explanations, would help to: identify what needed to be explained, determine the depth and complexity of the explanation, provide knowledge to aid the user in forming his or her own explanations, and suggest the system behaviour which would best assist the user in achieving his or her goals, as well as indicating appropriate "low-level" aspects such as the vocabulary to be used, the mode of address and so on. There is no doubt that if it were possible to acquire reliably the necessary knowledge about the user, a user model could be very valuable for explanation.

Nevertheless, as I have suggested elsewhere (Gilbert, 1987c), while systems should deploy what information about the user they can, a real user *model*, which simulates a user's interests and goals and which is based on accurate and specific data about a particular user, is a very long way from being realized. Two main forms of user model have been proposed: the "stereotype" models which attempt to represent the characteristics of a range of generic users (for example, a distinction is often made between "naive" and "expert" users) and "intentional" models which try to model users' plans, beliefs and goals. The problem with the "stereotype" models is that typical human–computer interactions provide very little data useful for choosing between the stereotypes. This means that one cannot provide a great number of finely graduated stereotypes and hope that the program will be able to select between them; the data will only warrant choice amongst a few gross characterizations and these will be insufficiently detailed to allow explanations to be as carefully tuned to the user as one would want.

The more ambitious "intentional" models are designed under the assumption that users have well-defined and stable plans and goals. The models attempt to discover these plans and goals from the user's dialogue with the system. They then use this knowledge to predict the user's behaviour. However, the initial assumption, that users are rule and plan guided, is probably incorrect most of the time. Although users may be able to offer plan-based accounts of their actions retrospectively, this does not mean that they routinely formulate well defined goals, devise plans and carry them out (Suchman, 1988). Systems based on user models which assume that users do behave in this way will obviously come to grief sooner or later in the interaction.

Some systems, in particular, intelligent tutoring systems, do need a model of the user (or the student) in order to function correctly. In the best tutoring systems, however, the user model really is a model, that is, it attempts to represent not merely the knowledge which the user is believed to possess, but also the process by which this knowledge is acquired and the "bugs" which may be included in it. A good example of such an approach to user modelling is that taken in some intelligent computer-aided instruction programs, where the student is modelled by a machine learning system. However, this is not the usual approach.

8 Dialogue

The debate about user models has generally taken as an assumption that the system's interaction with the user could be characterized as, "the user asks for an explanation and the system provides one". Because there is only one opportunity in this scenario for the system to generate its explanation, the output needs to be as perfectly formed as possible. Explanation is conceived as a one-shot, all or nothing response to a single stimulus from the user.

An alternative is to make the relationship between user and system truly interactive, with the two parties engaged in a sometimes extended dialogue (Moore and Swartout, 1988). This approach has a number of important implications. First, the system must be capable of maintaining a coherent and "natural" dialogue with the user and this affects the design of the interface. Second, the system has an opportunity to learn about the user during the dialogue and, in particular, to improve its understanding of what explanation the user needs. This means that explanations can be tried out and improved in relevance and comprehensibility, in conjunction with the user. Third, it may be possible to involve the user in the explanation to the point where the explanation is better seen as jointly constructed and emergent from the dialogue, rather than the sole responsibility of the system (Gilbert et al., 1989).

A major advantage of explaining through dialogue is that a strict division between roles where one party asks a question and the other provides an explanation is no longer necessary. Much more complex interaction structures are possible. For example, the explainer may wish to ask the explainee whether a developing explanation is relevant to the explainee's requirements. And the explainee can tell the explainer what he or she already knows, so focusing the explainer's attention on to the puzzling matters for which an explanation is sought. There are advantages, therefore, in permitting a richer sense of "interactivity" by providing facilities in the interface to allow both the explainer and the explainee to ask questions, provide answers and make statements.

9 The structure of dialogue

If both participants are given these facilities, however, a control problem immediately arises. How can one ensure that questions are answered and that the answers can be related back to the questions, if both parties can ask questions and make assertions at any time? One obvious solution to the control problem is to plan the dialogue in advance, perhaps using a schema or template which aims to guarantee a well formed and helpful interaction. Attempts have been made to identify such schemas from transcripts of people providing advice and explanation (Gilbert, 1987b; Rogers, 1988a). Although, as noted above, schemas are useful for planning individual turns of talk (McKeown, 1985), especially in giving explanations where clarity and good organization of the material is at a premium (Atkinson and Drew, 1979), it is less certain that a similar approach is effective on a larger scale to determine the structure of whole dialogues.

One reason for this is that pre-planned dialogue structures have, inevitably, to be rather inflexible, yet a characteristic of conversations is their contingent and casual nature. Neither party can precisely anticipate the other's next utterance: neither its content, nor even its timing in relation to the current speaker's talk. Hence, although a pre-planned dialogue structure could be used with the human participant forced to adapt to the system's controlling moves, the result will seem stilted, unnatural and no better than one expects of computers.

The alternative is to model the means of controlling the dialogue on the way people are supposed to do it. Work on "conversation analysis" (Levinson, 1983) suggests that, rather than using a global planning approach, dialogue should be 'locally managed". The implication is that what is said next depends on the immediate context, so that the dialogue organization operates on a turn-by-turn basis. In particular, such an organization does not depend on anticipating or even monitoring the overall shape of the dialogue.

How do people manage to interleave utterances so that there is no or minimal overlap without any obvious signal analogous to passing the baton from one speaker to another? The proposed solution, a good example of local management, is that there is a set of rules governing turn-taking to which participants orient. Utterances are perceived by hearers to be divided into "turn-ups", commensurate with syntactic structure such as clauses, noun phrases, etc. At the end of each unit, at junctures called Transition Relevance Places or TRPs, there is an opportunity either for the speaker explicitly to select the hearer to take the next turn (for example, by asking a question) or for the *hearer* to self-select; only if the hearer does not take over the floor is the speaker permitted to continue (the general turn-taking rules are more complex than I have indicated: see Sacks, Schegloff and Jefferson, 1974). Such a mechanism has a variety of interesting features: it makes it possible to identify "attributable" silence in conversation (as contrasted with a merely unintended silence); it provides an intrinsic motivation for hearers to attend to speakers in order to listen for TRPs; and it is able to function even when the parties cannot see each other to monitor gaze, thus explaining how it is possible to hold fluent conversations over the telephone.

In the context of explanatory dialogues, the rules for turn-taking proposed by conversation analysts imply a quite different interface from that provided on most computer systems (Frohlich and Luff, 1989). Typically, the system asks the user a question and then waits indefinitely for an answer. While the system is in the course of displaying the question, the user is unable to interrupt. In contrast, according to the Sacks, Schegloff and Jefferson rules, the user should be given an opportunity to "speak" after every TRP, or roughly, at the end of every clause. This means that some means for the user to self-select needs to be designed into the interface. The user should reciprocate in allowing the system to take the floor after every user's TRP.

Another example of local management which is clearly visible in ordinary conversation is the "adjacency pair", a paired utterance such as question–answer, greeting–greeting and offer–acceptance. The first part of such a pair raises an expectation that the other participant will provide the second part. Adjacency pairs may be nested, producing "insertion sequences". If Q is a question and A an answer, a typical insertion sequence would be:

X: Q_1
Y: Q_2
X: A_2
Y: A_1

In such sequences, the relation between each question and its corresponding answer is maintained by the expectations set up by the first part. The inserted pair (or pairs, for the embedding can be quite deep) is restricted in content to the clarification and elaboration of the issues raised in the initial question (Q_1).

An issue of some importance in conversation analysis which has significant implications for the construction of explanations is the focus on "breakdown" and "repair" (McTear, 1985). Not everything always goes well in conversation: it is not uncommon for the participants to mishear or misunderstand each other. However, troubles of this kind are not crippling because there are conversational mechanisms which will allow speakers to repair the problem. For example, there are "side sequences" (Jefferson, 1974) where time is taken out of the main flow of conversation to repair some trouble:

X: S_1
Y: What?
                X: S_2
X: S_3

Here, Y's "What?" signals a problem with X's statement, S_1, which X repairs with S_2 before continuing on the main track with S_3. This is one simple organization of breakdown and repair. There are several others.

Repair mechanisms depend on a number of features of conversation. First, there must be a way for speakers and hearers to monitor success and failure (the latter demanding repair). The fact that utterances take recognizance of prior utterances (for example, answers orienting to prior questions) provides one such way, as when a question fails to receive a reply hearable as an answer. Second, there must be a mechanism for doing the repair which hearers can link back to the breakdown. Third, there must be a way of resuming the normal interaction. Side sequences of the kind illustrated above are a simple way of handling all three problems.

While conversation analysts have largely, though not exclusively (e.g. Sacks, 1972), been concerned with the formal structure of dialogue, one can apply the notions of local management, breakdown and repair to determining what should be said as well as how one should say it.

10 Explanation strategy

As Grice (1975) pointed out, hearers and speakers work on the assumption that conversational participants will follow the "Cooperative Principle", that is, they obey the four maxims of Quality (Be helpful), Quantity (Don't Waffle), Relation (Stick to the point) and Manner (Be Clear) (Stenton, 1987). In practice, this means that hearers are able to make inferences beyond what speakers have actually said. For instance, in the absence of contrary evidence, hearers will infer that the speaker's utterances truly reflect the speaker's beliefs and that they contain all the information which the speaker thinks is relevant, and nothing which is not relevant.

The cooperative principle facilitates an essential economy of speech (cf. the "experiments" reported in Garfinkel, 1967), tu it also provides a powerful way for hearers to gauge speakers' beliefs about the hearer without the need for them to be explicitly discussed. As noted earlier, explanations are best when they are designed for the recipient, that is, when their content is turned to the needs of the explainee. Such tuning depends on the explainer coming to have a grasp of the explainee's pre-existing knowledge and beliefs.

One way in which the explainer can gauge the knowledge of the explainee is, of course, to ask. (Goguen et al., 1983: 531) presents an example from a transcript in which a tax advisor explicitly requests background information in order to assess the explainee's knowledge about the tax implications of capital gains. But, as they observe, such an explicit enquiry is only to be expected if the explainee is not well known to the explainer and the topic is relatively arcane. An alternative way of estimating the explainee's knowledge is by drawing inferences from the dialogue using the cooperative principle.

Even if the explainer initially knows nothing at all specific about the explainee, as when two strangers meet, he or she can justifiably presume that the explainee holds a "commonsense" set of beliefs, such as "anyone" might hold. In addition to this basic set, the explainer may be able to infer from the context of the dialogue and the social role of the explainee that additional beliefs can be presumed. Thus the explainer may reasonably be able to assume a wide range of beliefs on the part of the explainee before the interaction has begun. However, these default beliefs about the knowledge of the other are tentative and revisable and may be amended as the dialogue proceeds.

The explainer's beliefs about the explainee can be used together with the cooperative principle to design the explanation. For instance, the explainer will not explain things which the explainee already knows. The explainee, on the other hand, will be able to assess, and if necessary, correct the explainer's beliefs about the explainee's knowledge by attending to what is included and excluded from the explanation (Joshi, 1982).

To see what this would mean in practice, consider a very simple example. Suppose that participant X (a person or a program) "knows" propositions A, B, C and D (i.e. believes these propositions to be true), and believes that the other participant, Y, "knows" A, B, C and N. Note that this implies that X believes proposition N to be false (for otherwise X would know N as well as A, B, C and D). Y knows propositions B, C, D and E and believes that X knows B, C and M (see Figure 2).

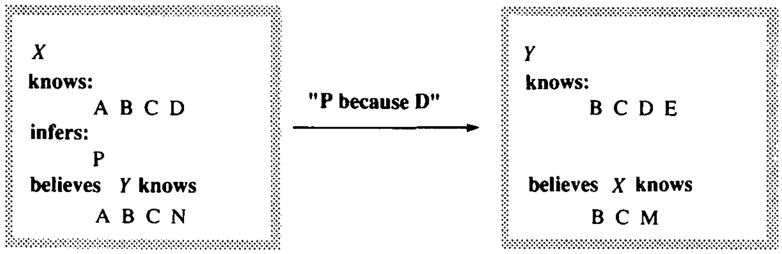

Figure 2

If X now infers proposition P from its knowledge of A, B, C and D, it should tell Y not only P, but also "spontaneously" explain P in terms of D. There is no requirement to refer to A, B or C because X believes that Y already knows these propositions. D, however, should be mentioned because X believes that Y is ignorant of D.

Y, however, already knows D and can infer that X's beliefs about Y's knowledge are incorrect, specifically that X believes Y does not know D when in fact it does. Despite being given an explanation, Y cannot infer P because Y does not know A, and so also infers that X knows something that Y does not.

Although Y cannot immediately infer P because it is ignorant of A, Y may be able to fill in the missing link by abduction (O'Malley, 1987a), that is, hypothesize that A may be true because P is. If so, Y can state, to X as a "checking move" (Ringle and Bruce, 1982) that "Oh yes, P because A". This indicates to X that X's belief that Y knows A is in need of revision. There is no requirement for X to confirm Y's statement explicitly since it is correct. X can infer from Y's mention of A that A was not in fact known to Y, but now is, and the dialogue can continue with other matters.

If, however, Y cannot abduce A, Y will have to seek clarification from X, asking "Why P?". The explanation offered by X should then mention A or C or both, because X knows only that its beliefs about Y's knowledge are wrong (for otherwise clarification would not be required) and cannot from the available evidence, discover whether the problem lies in Y's ignorance of A and C.

This is a rather simple example, particularly in that it assumes that both parties use identical deductive strategies and that their respective knowledge bases are each so small that they can be exhaustively searched for possible inferences (the consequences of relaxing the latter assumption are explored in Sperber and Wilson, 1986). Neither assumption is likely to be true of human–human dialogues. Nevertheless the example does indicate some of the interesting features and some of the power of dialogues. It shows the role of beliefs about the knowledge of the other in motivating spontaneous explanations (Sanmugasunderam et al., 1988). It demonstrates how each party's knowledge of the other can increase during the course of a dialogue. It also suggests that explaining can take place even in the face of inaccurate beliefs about the other, provided that those beliefs are revisable. Such revisions occur partly as a result of inferences made indirectly from the other party's utterances and partly from explicit requests for information and clarification.

Because beliefs about the other are revisable, and because mistaken beliefs are visible to the other (through stating what is already known or failing to state what needs to be said), it is possible to take remedial action when problems occur. Moreover, because breakdown is noticeable and repair possible, one effective strategy in dialogue when the other's knowledge state is not known is simply to presume that the hearer already knows everything needed for understanding. If the presumption is not true, the hearer can always ask for the missing knowledge. The advantage of this strategy of "maximum presumption" is its economy: not only does the speaker not have to first cross-examine the other party about its state of knowledge, but the speaker does not even have to engage in building an explicit belief set about the other's knowledge. Introspection seems to confirm that (except in certain special contexts such as teaching) we do not concern ourselves with building elaborate belief sets about other people's likely knowledge—we just talk and hope for the best.

11 Emergent explanations

One version of the dialogue which would be generated from the interchange between *X* and *Y* in the previous section is:

> *X*: P because of D
> *Y*: Oh yes, P because of A

To an observer lacking detailed knowledge of the belief states of the parties, it would not be clear from this dialogue whether *X* is explaining P to *Y* or *Y* was explaining it to *X*, that is, whether *Y*'s remark is a checking move or *X* is proposing a hypothesis which is then confirmed by *Y*. It may be that the best way of regarding dialogues such as this one is not to try to assign the role of explainer to either *X* or *Y*, but to see the explanation as an emergent feature of the interaction, so that we might say that *X* and *Y* jointly construct an explanation for P.

In more realistic situations, where the knowledge of both the participants is extensive, more complex and longer dialogues are likely to be required in order to establish a mutual understanding. In these dialogues it is likely to be even less clear which of the two is the explainer and which the explainee. For example, the dialogue may involve the expression of new ideas which had not previously occurred to either before or may involve the synthesis of facts and concepts contributed by both (Garrod, 1987). Miyake (1986) traces the development of mutual understanding as two subjects try to explain to each other the mechanism of a sewing machine and observes that the process in that domain seems to proceed through a series of steps, in which some incomplete level of understanding is reached and then found to be lacking, prompting further explanation.

As noted earlier, both parties will come to the interaction already possessing knowledge, without which it will be impossible for a useful interchange to take place. There is a possibility, however, that the pre-existing knowledge of the two parties may be in conflict, so that assertions offered by one are inconsistent with and, possibly, therefore incomprehensible to the other (in terms of the example of Figure 2, *X* "knows" Q while *Y* "knows" $\sim$Q). If the outcome is to be mutual understanding, one or other will have not only to acquire the other's knowledge, but also have to segregate and "unlearn" their previous knowledge. For instance, *X* may accept that its previous "knowledge" of Q was wrong before acquiring a belief in $\sim$Q, or *Y* may acquire Q at the expense of $\sim$Q. However, little is yet understood about the complexities of mutual understanding in the face of antagonistic belief systems and this must be an area for further research.

12 Conclusion

The initial reasons for getting knowledge-based systems to provide explanations were practical and pragmatic: explanations might improve the understanding and the confidence of the user, and, in the form of proof traces, they were easy to generate. Explanation was seen as little more than the icing on the cake of existing systems. However, as soon as one begins to consider explanation seriously in its own right, the inadequacy of this view becomes obvious. Furthermore, the intrinsically "social" nature of explanation—that to explain one has to involve two roles and that the explanation is often much more effective if the roles interact—has meant that explanation is increasingly becoming seen as jointly constructed through a dialogue and as emergent in the course of the encounter. For these reasons, we need to stop thinking of "explanation" as something which is generated by the computer on demand, and start thinking of it as a process which develops as a consequence of a symbiosis between the computer and the user.

A better understanding of this process will depend on work in a wide range of disciplines and specialties, as indeed research on explanation has always done. For example, cognitive psychology can clarify the ways in which two agents mutually develop a common vocabulary for talking about things that neither quite understands. Logic may help with problems of mutual belief. Advances in linguistics and knowledge representation are needed to improve text generation systems. Sociology can contribute to the understanding of social interactions between cooperating agents. And so on.

Happily just at the moment, interest in these and related issues seems to be on the increase and real progress is possible in the next few years.

Acknowledgement

This work was carried out as part of the DHSS Large Demonstrator project, supported by the Alvey Directorate of the UK Department of Trade and Industry and the UK Science and Engineering Council. The project collaborators were ICL, Logica, Imperial College London, and the Universities of Lancaster, Liverpool and Surrey. The stimulating environment provided by the other project members is gratefully acknowledged. The views expressed here are those of the author and may not necessarily be shared by the other collaborators.

An earlier version of this review was presented at the Fourth Workshop of the Alvey Explanation S.I.G. and I am grateful to the participants at this Workshop as well as to Betsy Cordingley for helpful comments.

References

Atkinson, JM and Drew, P, 1979. *Order in court*, London: Macmillan.

Cawsey, A, 1989. "Explanatory dialogues" *Interacting with computers* **1** 69–92.

Clancey, WJ, 1983. "The epistemology of a rule-based expert system: A framework for explanation" *Artificial Intelligence* **20** (3) 215–251.

Crossfield, L, 1986. "Explanation in regard to a welfare benefits advice system" *Proceedings of the First Explanation SIG*, Guildford. IEE.

DeJong, G, 1988. "An introduction to explanation-based learning" In: H. E. Shrobe (ed.), *Exploring Artificial Intelligence*, Morgan Kaufman.

Diaper, D, 1986. "Identifying the knowledge requirements of an expert system's natural language processing interface" In: Harrison, MD and Monk, AF (eds.), *People and computers: designing for usability*, Cambridge: Cambridge University Press.

Draper, S, 1987. "Spontaneous explanation" *Proceedings of the Third Explanation SIG*, pp 74–82, Guildford, IEE.

Frohlich, DF and Luff, P, 1989. "Conversational resources for situated action" *Proceedings of CHI '89*, pp 253–258, Austin, Texas.

Garfinkel, H, 1967. *Studies in ethnomethodology*, New York: Prentice-Hall.

Garrod, S, 1987. "Explanation in dialogue as attempts to overcome problems in coordination" *Proceedings of the Third Explanation SIG*, pp 83–93, Guildford, IEE.

Gilbert, GN, 1985. "Society, Judgement and Responsibility" *Seminar on the social implications of expert systems and artificial intelligence*, Abingdon, May 1985. British Computer Society Specialist Group on Expert Systems.

Gilbert, GN, 1987a. "Questions and answer types" In: Moralee, DS (ed.), *Research and development in expert systems IV*, Cambridge: Cambridge University Press, pp 162–72.

Gilbert, GN, 1987b. "Advice, discourse and explanations" *Proceedings of the Third Explanation SIG*, pp 94–109, Guildford, IEEE.

Gilbert, GN, 1987c. "Cognitive and social models of the user" In: Bullinger, H-J and Shackel, B (eds.), *Human-computer interaction—Interact '87*, Amsterdam: North-Holland, pp 165–172.

Gilbert, GN, Buckland, S, Cordingely, ES, Frohlich, D, Hardey, M, Jirotka, M, Luff, P and Robinson, P, 1989. "Providing advice through dialogue" *University of Surrey Social and Computer Sciences Research Report.*

Goguen, JA, Weiner, JL and Linde, C, 1983. "Reasoning and natural explanation" *International Journal of Man-Machine Studies* **19** 521–559.

Graesser, AC and Murachver, T, 1985. "Symbolic procedures of question answering" In Graesser, AC and Black, JB (eds.), *The Psychology of questions.* New Jersey: Lawrence Erlbaum, Hillsdale.

Grice, HP, 1975. "Logic and conversion" In: P. Cole and Morgan, J (eds.), *Syntax and Semantics*, New York: Academic Press, pp 41–58.

Guindon, R, 1988. 'A multidisciplinary perspective on dialogue structure in user-advisor dialogues" In: Guindon, R (ed.), *Cognitive Science and its applications for HCI*, New Jersey: Lawrence Erlbaum.

Harman, G, 1986. *Change in view*, Massachutts: MIT Press.

Hughes, S, 1986. "Question classification in rule-based systems" In: Bramer, M (ed.), *Research and development in expert systems III*, Cambridge: Cambridge University Press.

Jefferson, G, 1974. "Side sequences" In: Sudnow, D (ed.), *Studies in social interaction*, New York: Free Press.

Joshi, AK, 1982. "The role of mutual beliefs in question answering systems" In: Smith, N (ed.), *Mutual belief*, London: Academic Press.

Kidd, AL, 1985. "What do users ask? Some thoughts on diagnostic advice" In: Merry, M (ed.), *Expert systems '85*, Cambridge: Cambridge University Press.

Lehnert, WG, 1977. "Human and computational question-answering" *Cognitive Science* **1** 47–73.

Levinson, S, 1983. *Pragmatics*, Cambridge: Cambridge University Press.

Maybury, M, 1988. "Explanation rhetoric: the rhetorical progression of justifications" *Workshop on Explanation, AAAI '88*, pp 16–21, St Paul, MN, August, 1988.

McKeown, K, 1985. *Text generation*. Cambridge: Cambridge University Press.

McTear, M, 1985. "Breakdown and repair in naturally occurring conversation and human–computer dialogue" In: Gilbert, GN and Heath, C (eds.), *Social action and artificial intelligence*, Gower, Aldershot, pp 107–123.

Mitchell, TM, Keller, R and Kedar-Cabelli, S, 1986. "Explanation-based generalization: a unifying view" *Machine Learning* **1** (1) 47–80.

Miyake, N, 1986. "Constructive interaction and the iterative process of understanding" *Cognitive Science* **10** 151–171.

Moore, J and Swartout, WR 1988. "A reactive approach to explanation". *Workshop on Explanation, AAAI '88*, pp 91–94, St Paul, MN, August, 1988.

Nicolosi, E, Learning, MS and Bouroujerdi, M, 1988. "The development of an explanatory system using knowledge-based models" *Proceedings of the Fourth Explanation SIG*, pp 173–183, Manchester, IEE.

O'Malley, C, 1987a. "Understanding explanation" *Proceedings of the Third Explanation SIG*, pp 137–156, Guildford, IEE.

O'Malley, C, 1987b. "Explanation and expectation: why ask why?" *Proceedings of the Third Explanation SIG*, pp 157–167, Guildford, IEE.

Ortony, A and Partridge, D, 1987. "Surprisingness and expectation failure: what's the difference?" *Proceedings of the 10th IJCAI*, pp 106–108, Milan.

Pollack, ME, Hirchberg, J and Webber, BL, 1982. "Using participation in the reasoning process of expert systems" *Proceedings of the Second AAAI*, pp 358–361, Pittsburgh, PA.

Pratt, I, 1987. "Explanatory asymmetry" *Proceedings of the Third Explanation SIG*, pp 168–182, Guildford, September, 1987.

Ringle, MH and Bruce, BC, 1982. "Conversation failure" In: Lehnert, WG and Ringle, MH (eds.), *Strategies in natural language processing*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Robinson, P, 1988. "Explaining the benefit system" *Proceedings of the Fourth Explanation SIG*, pp 148–159, Manchester, IEE.

Rogers, Y, 1988a. "Feature analysis of explanations in a variety of task domains" *Proceedings of the Fourth Explanation SIG*, Manchester, IEE.

Rogers, Y, 1988b. Personal communication.

Rousset, MC and Safar, B, 1987. "Negative and positive explanations in expert systems" *Applied Artificial Intelligence* **1** 25–38.

Sacks, H, 1972. "On the analysability of stories by children" In Turner, R (ed.), *Ethnomethodology*, London: Penguin, pp 216–232.

Sacks, H, Schegloff, E and Jefferson, G, 1974. "A simplest systematics for turn-taking in conversation" *Language* **50** 696–735.

Sanmugasunderam, A, Spencer, B and Cohen, R, 1988. "The ThUMS Project: implementing user specific responses to an educational diagnosis system" *Research Report CS-88-25*, University of Waterloo.

Scott, AC, Clancey, WJ, Davis, R and Shortliffe, EH, 1977. "Explanation capabilities of production-based consultation systems" *American Journal of Computational Linguistics* **62** 201–211.

Shortliffe, EH, 1976. *Computer-based medical consultations: MYCIN*, New York: Elsevier–North Holland.

Sperber, D and Wilson, D, 1986. *Relevance*, Oxford: Blackwell.

Stenton, SP, 1987. "Dialogue management for cooperative knowledge based systems" *Knowledge Engineering Review* **2** 99–122.

Suchman, L, 1988. *Plans and situated action*, Cambridge: Cambridge University Press.

Swartout, WR, 1977. "A digitalis therapy advisor with explanations" *Proceedings of the 5th IJCAI*, pp 819–826.

Swartout, WR, 1983. "XPLAIN: a system for creating and explaining expert consulting programs" *Artificial Intelligence* **21** 285–325.

Weiner, JL, 1980. "BLAH, a system which explains its reasoning" *Artificial Intelligence* **11** 197–223.