Int. J. Expert Systems with Applications,, 8(4): 445-462 (1995)

Cooperative problem solving and explanation

L. Karsenty

CNAM
Laboratoire d'Ergonomie
41, rue Gay Lussac
75005 Paris
France

P.J. Brezillon

LAFORIA, Box 169
University Paris 6
4, place Jussieu
75005 Paris Cedex 05
France

Abstract:

Recent studies have pointed out several limitations of expert systems regarding user needs, and have introduced the concepts of cooperation and joint cognitive systems in the focus of AI. While research on explanation generation by expert systems has been widely developed, there has been little consideration of explanation in relation to cooperative systems. Our aim is to elaborate a conceptual framework for studying explanation in cooperation. This work relies heavily on the study of human-human cooperative dialogues. We present our results according to two dimensions, namely, the relation between explanation and problem solving, and the explanation process. Finally, we discuss the implications of these results for the design of cooperative systems.

INTRODUCTION

Recent studies have pointed out several limitations of expert systems regarding user needs, and have introduced the concepts of cooperation and joint cognitive systems [Woods et al., 1990] in the focus of AI. Our own experience in expert system projects has led us to believe that systems that place the user in a passive role, cannot provide effective support [see also Carr, 1992].

This paper is concerned with how the framework of cooperation can modify explanation requirements in the context of expert systems. Because few cooperative systems are currently available, we led our investigation from studies of human-human cooperative dialogues rather than directly from studies of human-computer interaction. This paper presents the first step of our study. This step involves describing the explanation features in a cooperation paradigm, as they emerge from the study of human-human cooperative dialogues. The next step will involve the implementation of these results. Although we have not begun to work on this second step, we indicate some elements that will help, we believe, in achieving such a task.

The results of the analysis of cooperative dialogues are split in two parts: the first focuses on the relation between explanation and cooperative problem solving, the second, on the process of explanation, which is conceived of as a cooperative process itself.

As far as the relation between explanation and cooperation is concerned, two major effects of explanations on the cooperation are noticed: (i) explanations may modify the agents' representation of the problem, enhancing their level of shared understanding of the problem and, consequently, their possibilities of coordination, (ii) explanations may modify the agents' problem solving knowledge. As a consequence of these possible changes, the problem solving process may take another direction, unforeseeable before an explanation need emerges. Hence, we then claim that *explanation must be conceived of as part of a cooperative problem solving process*, and not just as a parallel phenomenon that does not modify the course of the reasoning.

We would like to thank two anonymous reviewers for their helpful comments and suggestions regarding an earlier version of this article.

If explanations partly determine the cooperation, we may also say that the context of the cooperation, particularly the tasks to be achieved and the users' knowledge, determine the explanation needs. An extensive analysis of the explanation needs in the context of database design dialogues supports this claim, and suggests how task goals and users' knowledge may affect these needs.

We then present a view of explanation as a *cooperative process*. Previous work on explanation has already pointed out the need for an active collaboration between the explainer and the explainee, but considers that (i) the aim of this collaboration is to construct some explanatory content that the machine alone can hardly plan in some cases, (ii) the machine possesses the "right" interpretation of its actions and proposals, and the user has difficulties (misconception, lack of knowledge, ...) in reaching this interpretation. Our work not only elaborates on the possibilities of collaboration that two agents bring into play, but also--and perhaps what is more important--presents the idea that the process of explanation may be viewed as a *negotiation of meaning*. This implies that we do not believe that there is on the one hand one "right" interpretation, and on the other hand, some "erroneous" interpretations. We prefer to say that there are different possible interpretations of the same message, where the validity of each is always founded on some specific contextual knowledge. The goal of an explanation process is then *to adjust both agents' context* until compatible interpretations may be found.

Section 1 reports an analysis of some limitations of expert systems when one tries to implement them in actual working settings, and outlines the need for human-computer cooperation. The more advanced research on explanation in the expert system paradigm is presented in section 2. This allows us to identify the specific features of explanation in the cooperation paradigm, and sets the scene for our studies of human-human cooperative dialogues. Section 3 presents a methodology for analyzing explanations in dialogues, and proposes a new notation for graphically representing cooperative dialogues that makes it easy to see at a glance the high level structure of a dialogue. The results of this analysis are then presented. We will finally conclude in section 4.

1 THE EXPERT SYSTEM PARADIGM

1.1 The user and the system in the expert system paradigm

The primary design focus has been to apply computational technology to develop a stand-alone expert system that offers some form of problem solution. This is one main reason why the design of expert system has focused on problem solving, not on the user's needs. This paradigm emphasized tool building over tool use [Woods et al., 1990].

Locus of control typically resides within the machine, and the design intention is for the human to act as the eyes and hands of the machine. For instance, the system directs the user to make observations about the device's behavior and to take measurements from internal test points. Indeed, the human acts as an interface between the expert system and its environment. A classical scenario of user-expert system interaction is: the user initiates a session; the expert system controls data gathering; the expert system offers a solution; the user may ask for “explanation” if some capability exists; and, the user accepts (acts on) or overrides the system's solution [Woods et al., 1990].

1.2 Inadequacy of expert systems for end-users

First expert systems have not fulfilled much of their early promise. There are several reasons for this: A wrong representation of expert system users and their needs; the system inability to solve unexpected problems; the assumption of a well-formulated statement of the user's problem and the expert system opacity. We discuss these points below.

1.2.1 SE users are not novice, but competent agents

One generally assumes that the end-user is a novice only because he or she needs the expert's help for solving some particular problems. Actually, several studies of help dialogues and our own experience in expert system projects lead us to prefer to view the users as competent agents. They may have a high level of expertise, although this level may be quite different from that of the expert from whom the knowledge in the expert system was acquired. This expertise allows them to make early judgments about a problem and even generate partial solutions [Pollack et al., 1982; Woods et al., 1990; Maïs & Giboin, 1989]. Thus, one may ask what are the competent users' needs?

Several studies have pointed out that a competent agent does not remain "passive" in detecting a problem. Instead, he or she rapidly tries to explain the problem and identify a solution. The agent's information needs begin at this point. A few studies of natural help interactions support this statement.

For example, a study of network design dialogues between an experienced designer and a series of less experienced designers indicates that very few requests for solution are uttered by the less experienced agents [Darses et al., 1993]. Instead, they often propose elements of solution and the expert react by assessing the proposal. Moreover, the authors observe that:

(i) Either solution developments or preventive statements may follow the expert's positive evaluations;

(ii) A justification, often followed by alternative proposals, always comes with the expert's negative evaluations.

Other results have been extracted from analysis of competent agents' needs in diagnosis tasks [Woods & Hollnagel, 1987]. These authors claim that good advisory interactions aid problem formulation, plan generation, help determine the right questions to ask, and how to look for or evaluate possible answers. Thus, an effective help facility for competent agents must be able to answer questions like: What would happen if x? what are side effects of x? how does x interact with y? what produces X? how to prevent X? what are the preconditions (requirements) and post-conditions for X (given X, what consequences occur)? [Woods et al., 1990; see also Kidd, 1985; Alvarez, 1992].

Competent agents' needs are not confined to a list of questions that must be addressed when designing the system. They also include the need for another form of interaction where the agent may actively participate in the problem solving process. Therefore, some authors prefer to view user-expert dialogues as a negotiation process rather than a consultation dialogue [Pollack et al., 1982]. In this sense, solving the user's problem is finding a mutually agreed solution instead of just finding a solution. Pollack et al. show why the dialogue could take the form of a negotiation: Often, people calling for advice "have preconceptions about what a solution to their problems involves or what constraints it must satisfy" (p. 361). Given an expert's advice, the caller may want to understand why it is better than the one s/he was expecting. S/he also may want to check if her/his constraints have been taken into account by the expert.

The outcomes of these studies do not lead to the conclusion that an helpful system does not have to generate solutions. Rather, they show that this capability addresses just one type of the users' needs, if they are conceived of as competent agents. Such users need help in finding ways to progress in the problem solving process or in assessing their solutions. Moreover, these studies highlight that interacting with an expert is different from a consultation.

This section outlines that users' activities would benefit from a cooperation with a computer. The next section argues that a computer would also benefit from the user's help to extend its field of application.

1.2.2 Inability to solve unexpected situations

One source of difficulty in a problem solving setting lies in the number of particular cases. It is not possible to plan a priori all the cases that may arise. Now, the machine facing unexpected problems is often unable to solve them because, generally, the system knowledge has been extracted from the handling of a set of specific cases. With their experience, users could intervene at this level, but this would imply that the problem must be solved by a cooperation between users and the machine.

For instance, consider a real-world device that is composed of several pieces of equipment, some of them being theoretically identical. The behavior of the identical pieces of equipment may be different, due to a difference in their construction (e.g., different adjustments of relative timing). Such a variability is inherent in equipment, and may be a problem for the machine that only knows the theoretical functioning, rather than the real one. It may detect trouble when there are only particular constraints that define two different contexts for the identical pieces of equipment. In contrast, the user who is in charge of the real-world system, often knows why two pieces of equipment behave differently, and may take the 'right' decision (Perrot et al. describes such a situation in an application for the French power system [Perrot et al., 1993].)

This inability of expert systems to deal with rare and difficult cases can undermine the credibility of the system [Keravnou & Washbrook, 1989]. But it can also place users in a frustrating position, because it is facing such cases that they really need help.

1.2.3 Solving a problem: Finding out the problem

When expert systems provide solutions, they do not always solve the user's problem. There are, at least, two reasons for this:

1. With advice-giving systems, users often fail to ask the most appropriate question because they do no know what information they need [Pollack, 1985; Belkin, 1988; Baker et al., 1993]. Pollack notices that human experts compensate for the lack of relevance of users' questions by inferring their plan. This is one piece of evidence that shows that the expert must find out the problem before solving it. Baker et al. observe that defining the user's problem is not just confined to this act of plan inference. It also implies a dialogue, where both agents negotiate the user's request [Baker et al., 1993].
2. Another reason why a system may produce irrelevant solutions can be seen from an investigation of real interactions with an expert system [Woods et al., 1990]. The study was aimed at supporting technicians in troubleshooting an electromechanical device. The authors observed that different technicians facing the same problem with the device, provided different symptom descriptions to the system. Moreover, some of these descriptions deviated from what the knowledge engineer, which was in this case an expert of the domain, would have expected. These discrepancies between the provided data and the expected data led the machine to be off-track.

These observations highlight that the human-machine pair can be more efficient if both agents share the same understanding of the problem. Observations also reveal that it is not realistic to think that any user will have the same perception of the situation to be handled as the expert system. The analysis of cooperative dialogues between humans in section 3 will show that the sharing of the problem understanding relies on the explanation process.

1.2.4 A double-task: Solving the problem and understanding the system

We have said previously that the expert system and the user (or the competent agent) may have different levels of expertise relying on different experiences. These discrepancies may result in different lines of reasoning, which in turn may imply sequences of system questions that are incoherent to the user [Keravnou & Washbrook, 1989]. One can relate this problem to observations of users failing in following instructions for using a copier [Suchman, 1987]. Some cases of impasses arose because of mismatches between the user's and the machine assumptions about current world state and objectives. Actually, "following instructions requires actively filling in gaps based on an understanding of the goals to be achieved and the structural and functional relationships of objects referred to in the instructions." [Woods et al., 1990, p. 145].

Users need to understand system actions in problem solving as well as the result of the system reasoning [Teach and Shortliffe, 1984]. When these explanations are too difficult to find, users may refuse to use the expert system because, contrary to their expectations, their working load increases.

The first attempts at reducing the opacity of expert systems involved extracting explanations of the system behavior from the record of the rules that were fired during one session. However, this solution appears more informative and adapted to the knowledge engineer than to the end-users. Indeed, the knowledge engineer often introduces some technical constraints to limit the action domain of the acquired knowledge. This is the case for the screening clauses in rule systems [Clancey, 1983]. Such constraints then reduce the comprehensibility of the reasoning trace, which becomes difficult to follow. Other factors explain the inefficiency of this solution for aiding the users' understanding [Moore & Swartout, 1990]: The set of questions taken into account is limited; The system cannot justify its actions; Users cannot provide any feedback that would help the system in finding an explanation adapted to their needs; The system usually cannot provide alternative explanations. These limitations have led to much of the research presented in section 2.

1.3 The need for human-computer cooperation

Most of the work in developing expert systems focuses on the knowledge needed for the problem solving, not on the wishes of end-users. A consequence is that end-users must adapt themselves to the expert-system behavior. However, users generally have a high level of expertise, even if this expertise is different from that of the domain expert. As a consequence, computational technology should be used, not to make or recommend solutions, but to help users in the process of reaching their decision [Carr, 1992; Woods et Roth, 1988].

There are two different approaches to avoid the problems that are related to the user-expert system interaction. In the first approach, one increases the explanatory capabilities of the expert system. However, the relationship between both agents remains the same. In the second approach, one modifies the user-system interaction and introduces the notion of human-computer cooperation.

Our research aims at integrating both approaches, using studies of explanations in expert systems for designing cooperative systems. Indeed, this integration is not a simple transfer of the results gained from the first approach to the cooperation paradigm. The reason is that the cooperation paradigm modifies some explanation characteristics.

Few cooperative systems have currently been implemented. Hence, it is not easy to address the explanation issue directly from a human-computer interaction perspective. Thus, we have investigated the relationship between explanation and cooperation from studies of human-human cooperative dialogues.

Two features of cooperation settings must be stressed: First, the *control is shared* by the machine and the user in joint cognitive systems. Thus, either the machine or the user plays the role of the information provider at each turn, the other providing a support role. Moreover, a *mutual dependency* characterizes their relation, because both agents attempt to reach a common goal. In other words, each agent must take the other's actions into account.

This paper addresses two questions:

(i) How the features of cooperation, particularly shared control and mutual dependency, work and can affect the explanations?

(ii) What are the main differences between the explanation requirements for the design of an expert system *versus* those for a cooperative system?

2 EXPLANATIONS IN THE EXPERT SYSTEM PARADIGM [1]

One difficulty with expert systems lies in their opacity, i.e., their inability to make clear their reasoning and to justify the solution to the user in a convincing manner. This is particularly crucial when one wants to use the expert system for training and tutorial purposes [Clancey, 1983]. These problems have motivated much of the research on explanation.

The first attempts at generating explanations mainly involved providing a trace of the system reasoning. However, it soon became clear that this solution was not satisfactory. Recent attempts have involved the knowledge bases with information about the domain, the task, the ability to communicate with the user. At the same time, one provides tailoring and dialogue mechanisms that make the system more sensitive to the user. These aspects are discussed below.

2.1 Making expert knowledge more explicit

Most of the expert knowledge is compiled. Such knowledge expresses under a condensed form a number of knowledge pieces and their organization. Facing this compiled knowledge, the user may need the missing pieces of information for understanding the system reasoning.

Clancey (1983) shows how the rules in a classical expert system such as MYCIN have different justifications: identification rules (for classifying an object), causal rules, common sense knowledge rules, and rules of the domain. These justifications, the structural knowledge (for instance hierarchic relations between rules), and the strategic knowledge (e.g., "If there are non-usual causes, treat them") need to be made explicit for improving the possibility of understanding and modification of a system [see also Swartout, 1983; Hasling et al., 1984; Chandrasekaran et al., 1989; Wick & Thompson, 1989].

2.2 Tailored explanations

Three issues must be considered in building a system that is able to tailor its explanations: What must we tailored? Which is the information to use and how to get it? Which tailoring technique to use? The studies described below present a representative sample of the different answers to these questions that can be found in the literature.

Adapting the degree of details to the users' knowledge

- Wallis and Shortliffe (1984) use a technique for generating explanations of causal chains, which are customized to the users' level of expertise and to the amount of information that they want. The user declares these two values, on a scale ranging from 1 to 10, and can change them during the dialogue. They associate a measure of complexity to each rule and concept in the knowledge base and a measure of importance of each concept. User's needs and the appropriate concepts or rules are then matched.
- XPLAIN tailors its answers to users by varying the number of steps included in an explanation depending on the type of user [Swartout, 1983]. Viewpoint markers are attached to the steps in a prototype method, indicating which type of user the step should be explained to.

Adapting the explanation types to the users' knowledge

- TAILOR adapts the explanations according to a user model with stereotypes and individual features [Paris, 1990]. From her study of encyclopedia, Paris finds that: (i) Objects have to be described in terms of subparts and properties for a knowledgeable user; (ii) The explanation should be focused on how the object works for a novice.
- Moore and Swartout (1990) employ a user model to choose between the several operators that can be candidates for achieving a discourse goal. For example, if the goal is to describe a

[1] We thank Béatrice Cahour for her help in making this state of the art.

concept, one can describe its attributes and its parts, or draw an analogy, or give examples, etc. The choice is then based on the dialogue history and on the user model.

Adapting the explanation types to the users' goals

- ADVISOR tailors its explanations to the user's goal [McKeown et al., 1985]. It selects a perspective (or viewpoint like: Process of meeting requirements; State model process; Semester scheduling process). It then provides justifications for advice in choosing courses and planning for students, by inferring the user's current goal, and adapts the content of the explanation to the perspective. The user's goal indexes the perspective, which indicates the relevant information to include in the justification of the advice.
- Van Beek also studies how the content of an explanation should change as a function of the goals of a discourse situation and of the higher domain goals of the user [Van Beek, 1987].

In the above research, the explanation is conceived as a text that has to be generated. In the research presented in the following section, the explanation becomes a *process*.

2.3 Explanation as an interactive process

Sometimes users do not understand the explanation that is generated by the system, or need another type of explanation, or judge this explanation as not being useful. In such situations, an alternative explanation must be presented to fit their needs. The machine can then interact with the user to produce a new explanation. As a consequence, the interaction control must be shared by the machine and the user to allow follow-up questions at least.

One weakness of explanation facilities is partly due to deficiencies in text planning and generation strategies. This may be compensated for by a reactive approach to explanation [Moore & Swartout, 1990]. According to this approach, the explanation generation relies on the user's feedback. This is an alternative to the weakness of the user model. Indeed, interacting with the user, the system may use incorrect assumptions about the user, and, consequently, unsuitable explanations may be provided.

One solution is to build explanations interactively [Cawsey, 1993; Lemaire & Safar, 1991]. The user should be able to interrupt for clarification, and the system should check the user's understanding.

Another solution is to acknowledge the impossibility for the machine to provide explanations that are tailored to the user, and let the user build the explanation that s/he wishes [Brézillon, 1992]. This idea is implemented in a program with some functions that allow the user to interrupt the machine. An interruption is possible at some points depending on the knowledge structures of the domain.

2.4 Conclusion

The main advances brought about by the studies quoted above are twofold:

- First, the knowledge that is used by the reasoning process encompasses contextual knowledge. One must make explicit this contextual knowledge for producing better explanations, especially when the user has a type of expertise that is different to that of the expert. This leads to a revision of methodologies of knowledge acquisition: The knowledge that is needed for explanation must be elicited with the knowledge for the problem solving [e.g., Dieng et al., 1992].
- Second, cooperative explanations may be produced by using a user model. But this is not always sufficient. The system may lack information on the user or even may have a wrong user's representation. As a consequence, it is important to allow the user to intervene in the explanation building.

These studies aim at making more acceptable expert systems. We believe that enhancing the explanatory power of computers is not sufficient to reach this aim. The main problem with expert systems lies in the passive role given to the user. Thus, we prefer to look for a more cooperative solution, one where both agents can articulate their specific abilities. However, choosing this option leads us to expect different features of the explanation. This is not only because in cooperative problem solving, there is not just one agent who is attempting to solve the problem at hand. It is also because cooperating agents maintain a specific kind of relation that one may call mutual dependency.

Two studies of cooperative dialogues were conducted to give an answer to the following question: How is an explanatory process affected by cooperative work? Actually, this question contains three sub-questions:
(1) What are the effects of an explanation process on the agents' tasks and reasoning?
(2) What is the relationship between explanations and the cooperation context?
(3) How can we model a cooperative process of explanation?

3 EXPLANATION IN THE COOPERATION PARADIGM

3.1 Data collected

We studied two types of cooperative dialogue that were recorded in natural working situations. We present each of these dialogues as well as the characteristics of each cooperation.

3.1.1 Validation dialogues

The first type of situation observed involves database validation dialogues. In this situation, a designer submits his solution--a database conceptual schema--to future users. The application domain is inventory control of garments. Four dialogues of this type were collected with the same designer and two different groups of future users.

This type of cooperation is aimed at providing the designer with the most knowledge supporting his solution. This cooperation relies mainly on the *complementarity of the agents'knowledge*. Future users possess domain knowledge (activities, information supports, organizational criteria, etc.) and the designer possesses design and database knowledge.

These dialogues present a common structure: The designer presents his proposals (each proposal corresponds to a component of the database conceptual schema) and an assessment process takes place. This process may lead to:
(i) A direct agreement by the future users without further information;
(ii) The users' agreements after the production of an explanation;
(iii) A modification of the initial proposal after a negotiation phase.

The validation dialogues were recorded and subsequently transcribed. The resulting verbal material constitutes the basis for the analysis.

3.1.2 Design dialogues

We have collected design dialogues during two real design projects in aerospace industry. A cooperation between an engineer and a draughtsman is the basis for the design in the organization where the study was conducted. This cooperation relies heavily on a *complementarity of the agents' competence.* Engineers have the competence to determine materials and measurements. Draughtsmen have the competence needed to determine shapes and principles of integration between the components of a mechanical device. The need for a cooperation arises for two main reasons: (i) The two levels of decision (materials, shapes, etc.) interact; (ii) Both designers' specific experiences are sometimes needed (e.g., to choose between two alternatives).

We followed two pairs of designers, each pair having the responsibility of one project during a period of 2 and 4 weeks. We videotaped most of the task-oriented interactions and subsequently transcribed the communications. The problems handled within the projects are different in many respects: (i) The history of the design is different: One project concerns the redesign of a previous solution due to a problem identified during some tests. The other concerns the adaptation of a previous solution to new requirements. (ii) The type of mechanical function to be achieved, the component and the structure of the device aimed at, and the organizational constraints were also different. Moreover, time and cost constraints are more salient in one project than in the other.

The form of the dialogues occurring in these situations differs from the form of the validation dialogues. Initially, the problem is stated (generally by the engineer) and a preliminary solution is cooperatively established. At the same time, both designers define the tasks to be accomplished. The subsequent meetings are of two types: (i) Pre-planned meetings aimed at assessing the solution state; (ii) "Opportunistic" meetings held each time one of the designers finds out a new problem with the current solution and cannot solve it alone.

The cooperation in the first project proceeds essentially with pre-planned meetings. In the second one, the project evolves with "opportunistic" meetings. Thus, data concerning the first project correspond to two dialogues (around 90 minutes each), while data concerning the second project correspond to eleven shorter dialogues (from 10 to 20 minutes).

Data gained from these dialogues contain communicative behaviors and other physical behaviors needed for achieving individual tasks. We retain some of these behaviors in our coding when they clarify verbal statements. They include retrieval of information in documents, drawing activities, and operations with CAD tools.

3.2 Methodology for analyzing explanations in dialogues

We now describe a basic model of cooperative design and the methodology of dialogue analysis that partly bears on it.

3.2.1 A model of cooperative design

At a very abstract level, individual design activity may be seen as an iterative process beginning with ill-defined requirements (or design goals) which lead to the generation of some preliminary solutions. The design project then elaborates on and assesses the solutions that are generated. This process generally leads to a better definition of the requirements, and, as a result, either the refinement or the substitution of the preliminary solutions that have been generated [Malhotra et al., 1980].

One may conceive of cooperative design as a group of agents that intervene in the following manner. One agent brings requirements, then either another one formulates a solution, or a joint action produces a solution (i.e., when different abilities are needed to produce a solution state). The members of the group assess the solution that is produced. Eventually, each member may add data that modify the design goal statement. The problem solving then continues at a deeper level of problem understanding.

Cooperative design is a process where each agent has his or her own competence that must be coordinated with the other agents' competence. Thus, other agents' agreement is necessary at each step of the process. This means that each proposal must be mutually agreed to be taken into account by a cooperation (see [Pollack et al., 1982]). For us, this statement is equivalent to saying that proposals must be *mutually explainable*.

3.2.2 Data analysis

The analysis of cooperative dialogues presented below is based on two ideas:

1. A dialogue contains two classes of communicative act: *Task-driven acts* and *acts of explaining*. Communicative acts within the first class convey the information that is required by the task at hand. Their definition may be independent of a particular interaction between agents. Acts of explaining convey information required to ensure the success of the task-driven acts. They are dependent on the agents' ability to understand each other[2].

An illocutionary act and a propositional content constitute each task-driven act [Searle, 1975]. In design dialogues, illocutionary acts are: Inform, propose, ask for information, ask for agreement, criticize. Propositional contents are: Goal, problem data, action, solution, assessing (negative or positive), meta-planning or interpretation (extracting data from blueprints, it is

[2] Ethnomethodologists have proposed a model of dialogue very close to this one, which accounts for repair [Jefferson, 1972]. Problems can lead to *side sequences*, that are oriented for providing a remedy to the problem, so that the *main sequence* of talk can continue. One may consider explanations as "side sequences" too, regarding to the main sequences that are composed by task-driven acts.

often necessary to find out what the drawing meaning is). One particular type of propositional content is termed "cognitive state". It occurs when one agent states (or asks) about his state (or the other's state) of knowledge concerning the considered topic. Examples of utterances coded in this way are shown in table 1. Moreover, we have achieved further coding of the content. It invovles the links between different statements of goals, actions, solutions or problem data. These links may be: refinement, correction, or alternative.

Table 1: Examples of utterances coded

You have the values concerning this part in J.C.'s doc.	*INFORM (Information source)*
What we have to do is to prevent the ring from rolling.	*PROPOSE (Goal)*
[To prevent the ring from rolling], we are going to achieve a higher tightening on the axle.	*PROPOSE (Solution)*
We will see later what measurements are necessary.	*PROPOSE ("Meta-planning")*
Do we have to take rigidity into account?	*ASK-INFORMATION (Goal)*
Do we begin to put the allowance values?	*ASK-AGREEMENT (Action)*
No, you don't need to do this.	*CRITICIZE (Action)*
This axle corresponds to the wheel axle.	*INFORM (Interpretation)*
Did you already see one similar device?	*ASK-INFORMATION (Cognitive State)*

2. The second idea on which the dialogue analysis is based is the following. One explanation (or act of explaining) conveys *contextual information* that is needed for understanding and accepting task-required information. This information is *missing from the explainee's context*[3]. In communication, context is the set of shared beliefs making an utterance relevant, and allowing the hearer to recognize the speaker's intended meaning (For a discussion about the notion of context and its use in the communication process, see Cahour & Karsenty, 1993, and Mittal & Paris, this volume.)

In some cases, it is sufficient to know the nature of the question that precedes an act of information for determining if a piece of information is or is not an explanation. In other cases, the occurrence of linguistic explanation marks (because, for, since, etc.) reflects largely the speaker's explanatory intent. However, explanations are not always given after questions or indicated by linguistic marks. Consequently, we need some criteria to decide if a given utterance is an explanation or not. Two types of criteria have been used:

• The first criterion concerns the relation between the information-to-be-explained and the information-which-explains. We state five general types of explanatory link: Definitional, causal, end-mean (or intentional), functioning, and justificative. One expresses (and tests) each type of explanatory link as a question linking two utterances. Relationship between explanatory type and question are:

- *Definitional*: A question like "What is X?"
- *Causal*: A question like "Why X?" . Moreover, the proposition-to-be-explained must be seen as an effect of the proposition-which-explains.
- *End-mean* (or intentional): A question like "What X for?"
- *Functioning*: A question like "How X works?"
- *Justificative*: A question like "Why X, and not Y?"

It is important to note that the identification of explanatory links requires domain knowledge, especially when no linguistic mark indicates the presence of an explanation. The following example illustrates this point (A and B are two agents, and the number classifies the corresponding utterance):

A1 What is this finger for?
B1 For preventing the rolling.
B2 You have the tightening, and a finger inside the rotation.

[3] Some research experimentally demonstrated that this characteristic is central when choosing the explanatory information [see Turnbull & Slugoski, 1988].

Actually, the utterance B2 attempts to explain how the finger (a component of the mechanical device) can prevent the rolling of a ring. The analyst can perceive the explanatory nature of the utterance only if he has knowledge about how the function described in B1 is technically implemented. Otherwise, it is difficult to identify the explanatory link between B1 and B2. However, in some cases, the analyst could assume an explanatory link even with a lack of knowledge (because of the intonation for instance). When such a situation arose, we always asked the designers participating in the dialogue for confirmation of the type of explanatory link suspected.

- The second criteria used for the identification of explanation comes from the fact that explanations must convey *new pieces of information* assumed to be unknown by the explainee, or at least not present in his mind. As a consequence, one acknowledges an explanation only if it is possible to state that the explainer thinks that the explainee missed the conveyed information. This may appear directly in the dialogue. For instance, when the speaker says "We have changed the form of this part because a new problem arose …,". Or this may be possible because of the analyst's knowledge of the project. If the analyst has followed all the previous meetings between the cooperating agents, s/he is often able to identify new pieces of information.

3.3.3 Graphical representation of cooperative dialogues

We present a visual representation of dialogues to comprehend better (and communicate) some characteristics of explanation in cooperative dialogues. Figure 1 provides an example of the representation used. Note that explanations appear on the lower part of each line. Note also that agreement requests are indirectly coded by specifying when an agreement is produced as a response (showed by the sign "*".) We divide the course of dialogue in phases according to the nature of the propositional contents. Even if this cut-out sometimes relied on discourse cues (for instance, the passing to an execution phase is reflected by an utterance of the kind: "OK, well, we try to do it?"), often it was the result of the analyst's subjective decision.

[Insert figure 1 at this level in a separate page]

This type of representation has several advantages for the study of explanation in dialogues. It allows us to directly notice when and how many explanations are produced. It gives a rather abstract view of explanatory dialogues (the process of explanation). Besides these advantages, it also permits us to see very easily which role is taken by each agent in each phase of the dialogue, how many questions were asked and by whom, how many disagreements have been encountered, etc.

3.3 Explanations for cooperating

3.3.1 Bilateral explanations

In design dialogues, each agent needs the other's competence and knowledge. Consequently, each agent can make a proposal (of action, solution, assessing and so on), and may need to explain her/his point of view. The representation of a phase of solution negotiation given in Figure 1 illustrates this point. Each agent produces a negative assessment of a solution that has been previously proposed, and explained his other judgment.

The need for bilateral explanations points out a primary difference between cooperation situations and interactions with traditional expert systems. If we break the asymmetry of knowledge that is imposed by the expert system paradigm (user as novice, machine as expert), the explanation problematics involves more than making systems able to generate ("good") explanations. The general issue becomes how to allow agents to cooperate better by giving them the ability to explain themselves and understand the other's explanations.

Very few studies attempt to provide users with the possibility of formulating their own explanations to the system. Obviously, the main reason is the system limitations for understanding users' utterances. However, researchers attempt to bypass this problem by allowing users to communicate with the machine in a more formal language (e.g., see [Coombs & Alty, 1984]).

3.3.2 Quantitative significance of explanation in cooperative design dialogues

The rate of explanations per dialogue is the number of words included in each type of communicative act (explanation versus others).

Globally, the explanation portions correspond to 31.5% of the entire content of the dialogues collected. This rate varies according to the projects followed: 28.4 % for the first one, 34.3 % for the other. Because each project differed from the other in many respects (designers' characteristics, type of problem, organizational constraints as time, etc.), it is not possible to explain these differences.

This result shows the significance of explanations when two people with different background knowledge and (partly) different goals want to work together to reach a common goal. However, it is difficult to estimate the generality of such a result, partly because only two projects were examined, and partly because few other studies achieve a similar account of explanation in dialogues. The relevant study that is similar to ours, focused on causal explanations occurring during everyday conversations [Nisbett & Ross, 1980]. The authors found that statements expressing or requesting causal explanations account for approximately 15 % of all utterances.

3.3.3 Relations between problem solving and explanation

Given a problem, each agent in a cooperation can elaborate goals, or plan actions, and at least partly elaborate on and assess the solution. An optimal cooperation would be one where each piece of information that is provided by one agent, would be directly taken into account by the other to produce the next step of the problem solving process. In some domains, where the situations are all predictable and the procedure well-defined and well-rehearsed (e.g., aviation), cooperation proceeds in most cases such as this. As a consequence, one may note that the language that is used is very restricted and explanation needs occur rarely [Falzon, 1991].

It is well known that the features of design activities are exactly the opposite of these ones: The situation, or the problem state, and the goals are ill-defined, and there is no well-defined procedure to reach these goals. This is the difficulty for predicting the next step in a design activity. However, according to the model of design described above, we could expect the following sequences of communicative act:

- [A expresses a goal, B agrees and formulates relevant problem data, an action or a solution proposal]
- [A proposes a solution, B expresses a solution assessing, or directly elaborates on it]
- [A proposes an action (optionally, with data to be taken into account); B expresses an action assessing, elaborates on it, or directly executes the action]

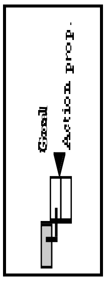
We call "logical" chaining of actions such sequences. The logical aspect refers to the model of design activity that is described above. Following this, we could ask: "What are the effects of explanation on the logical chaining of actions?" Two main effects of explanation have been noticed:

1. Explanation may be a condition to go to the next action

Three kinds of observation support this statement:

A) When one gives an explanation before the information-to-be-explained (the target), the agreement usually follows the target utterance, and concerns at the same time the explanation and the target utterance. There is not a special agreement for the explanation. Moreover, the next "logical" action follows from the agreement.

[Example 1] (in the extracts presented below, E refers to an engineer, and D refers to a draughtsman)

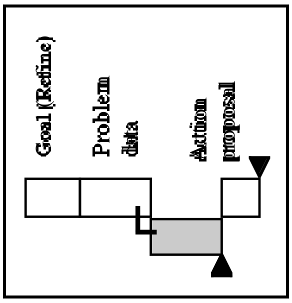


E I have no dimension here (*pointing to the picture*). **[JUSTIFICATION]**
So, you're forced to put the dimensions of the components that are beside it. **[GOAL]**

D Ok. **[AGREEMENT]**

E For beginning, you can take the dimensions of the bracket directly. **[ACTION PROPOSAL]**

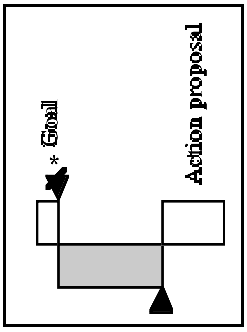
B) When a proposal is provided with an explanation, most of the time, the agreement follows the explanation, and the next "logical" action occurs.
[Example 2]



E Well, we need to determine the clearance which is necessary to avoid a contact. **[GOAL]**
X had achieved a kinematics study, which is here (pointing out a document) **[PROBLEM DATA]**
I With his CAD tools, he turned the actuator, and he noticed what dimensions were staying between the two components. **[EXPLANATION]**
E mmh mmh **[AGREEMENT]**
I Well, the best would be to calculate the order by hand to determine all these dimensions. **[ACTION PROPOSAL]**
E mmh, OK. **[AGREEMENT]**

We note that in a few cases, the hearer may agree on the explanation content, but disagree on the explanatory relation. Generally, this situation causes an extension of the initial explanation.

C) When a disagreement is expressed, the "logical" chaining of actions stops momentarily. In some cases, the explanation given by the agent by whom the proposal had been rejected, modifies the other's judgement and, as a result, allows the next "logical" action to appear.
[Example 3]



E We need to determine the surface-envelope of the trajectory of this point. **[GOAL]**
Is it OK? You see how … **[ASK-AGREEMENT]**
D No, I didn't … **[DISAGREEMENT]**
E You have the axle here (*show*), the bracket and the actuator that is below. The objective is to allow the rotation. The contact… you know that you can get a contact of this type (*shows with his hands*), but it will be going to move, one time here, one time there, according to the actuator position. **[EXPLANATION]**
D OK, OK, … **[AGREEMENT]**
E Well. So, I will say: We put on the actuator a track of the contact. **[ACTION PROP.]**

2. Explanation may modify the direction taken by the cooperation.

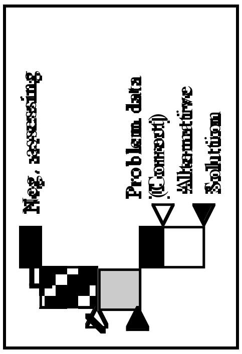
Once the agents reach the next step in the problem solving, they also restrict the set of potential following steps. If a proposal is accepted by others, it will affect the subsequent actions of the cooperation. For instance, if a goal G is accepted, the other alternatives are rejected. The subsequent solutions that will be generated depend on the earlier solution.

We observe that sometimes a communicative act that is refused, leads to a next step that did not fit the "logical" chaining of actions described above. For instance, we have observed the following kind of sequence:

Solution proposal ->Disagreement -> Explanation -> Agreement
-> Alternative Solution -> Agreement

We call here 'explanation' an item that corresponds most of the time to a *negotiation* during which each agent attempts to explain his orher disagreement regarding the other's view. The act of explaining leads them to *make their context explicit*, i.e., to formulate the set of beliefs that establishes their position on the topic. This gives them the opportunity to discover eventual discrepancies between either their problem statement or their underlying knowledge. There are some cases where the direction of the cooperation is modified after the provision of the explanation. This exhibits situations where the first speaker had either a wrong representation of the problem (see [example 4]) or lacked domain knowledge (see [example 5]). By changing either one, the explanation and negotiation process may influence the way agents progress in problem solving (one may find a similar conclusion in Cawsey et al., 1992).

[Example 4]



D No, we're going to be worried with this. **[NEGATIVE ASSESSING]**
Look: you've 50 mm here, and there you've a flat surface which is like that. We will put a part, mmh … **[EXPLANATION]**

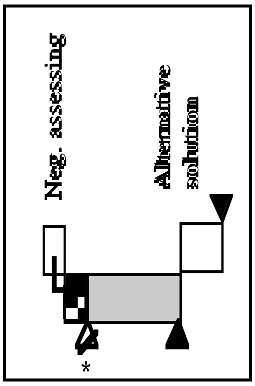
E No. Because from the angle viewpoint, it is this one that is crossing. **[DISAGREEMENT + EXPLANATION]**

D …OK. Then, the bracket does not matter anymore. **[AGREEMENT + CORRECT: PROBLEM DATA]**

E Right. … This means that we'll need to take some measurements on this part directly, and we'll make the corresponding adjustments. **[AGREEMENT + ALTERNATIVE SOLUTION]**

D Yes, right. **[AGREEMENT]**

[Example 5]



E We've to avoid that. **[NEG. ASSESSING]**

D Why ? It does not hold ? **[ASK-AGREEMENT: CAUSAL EXPLANATION]**

E That's not the point. **[DISAGREEMENT]**
Actually, that provokes significant contacts which result in a rolling of the actuator that is not perfect. … **[EXPLANATION]**

D Yeah, OK. **[AGREEMENT]**

E Therefore, the solution consists in determining a complex surface to get a contact cleara **[ALTERNATIVE SOLUTION]**

D Yeah. **[AGREEMENT]**

During an explanation process that follows disagreement statements, agents tend to *adjust their context* until they reach a compatible interpretation of the proposal. There are two side-effects of this adjustement process. First, the shared understanding of the problem increases. As a consequence, we may expect that the cooperation will be more efficient for handling the current problem because it is better coordinated. Second, each agent's shared knowledge increases. As a consequence, we may expect that the cooperation will be more efficient for solving future problems.

Discussion

These observations on the relation between problem solving and explanation point out a second difference between cooperation and expert system paradigms. An expert system is like an "oracle.", i.e., it is able to find the "right" solution. Explanations are then necessary *to convince users* of the validity of the system results. As a consequence, designers of expert systems usually have not considered the effects of explanation on the problem solving process. Within the cooperation paradigm, explanations are a *part of the problem solving* because solving a problem cooperatively is trying to reach a mutually agreed or mutually explainable solution.

The observations that are described above highlight another specific feature of cooperation: The relation between *knowledge acquisition* and cooperative problem solving. According to these observations, we claim that each disagreement between agents may be a timeliness for acquiring new knowledge. Moreover, we notice that, in a cooperation, knowledge acquisition relies heavily on the confrontation between each one's explanations.

3.3.4 Explanation needs and context

We extract the results discussed in this part from the study of database validation dialogues. Most of the explanations that are observed in these dialogues are designers' *spontaneous explanations*. The summary of explanation needs that is presented below (table 2), is based on the analysis of these spontaneous explanations. This analysis shows that explanation needs depend, at the same time, on the task, the type of information conveyed and the users' knowledge giving a meaning to this information.

Table 2: Explanation needs in database validation dialogues[4]

1. Definitional:
    - What is X?
    - What is the difference between X and X'?
2. Design rationale
    - 2.1 *Existence*:
        - Is X true? (Here, X is a relationship. For instance: Employees can_buy Garments in_a Catalog')
    - 2.2 *Necessity*:
        - What is X for?
    - 2.3 *Requirements rationale*:
        - What caused the existence of a given need?
    - 2.4 *Consequences*:
        - What are the consequences of X?
        - What are the advantages of X over X'?
        - What if not X?
3. Database use
    - 3.1 *Procedure and context of use*:
        - When and how to use X?
        - How is G reached?
        - What is the need to use X?
    - 3.2 *Justification of a procedure of use*:
        - Why do we need P?
    - 3.3 *Consequences of a procedure*:
        - What are the consequences of P?
        - What are the advantages of P over P'?
        - What if not P?

Explanations are of three general types in the database validation dialogues: definitional, design rationale and database use. Each of the two last types is composed of subtypes. Each subtype is defined below by one or many questions that it represents.

The account for explanation types in database validation dialogues confirms previous studies showing that users' questions addressed in traditional expert systems literature do not reflect the end-users' actual needs (e.g., see Kidd, 1985; Hughes, 1986; Gilbert, 1987).

Moreover, it is important to note that the target of these questions is never the part of the designer's reasoning. This reasoning part concerns his own expertise, the way by which the domain may be structured into a conceptual schema. Thus, the designer never explains, for instance, why he had chosen to structure the employee's addresses as an attribute rather than an entity.

At the same time, we notice that when database designers discussed a given conceptual schema, the information structure was a central concern of their discussion. Indeed, one of their main objectives was to be ensured that the structure of the data proposed was the best, given technical criteria such as data integrity, rapidity of access, etc.

[4] 'X' generally represents one datum, as the employee's name, 'P' refers to a procedure, or sometimes a single action, implied by the use of the future database, and G refers to a users' goal.

Discussion: Explanation needs, perspectives and goal-oriented interpretation

This difference between end-users and database designers highlights that the same information presented in the same context of task (the validation of a solution) could result in different goals and, as a consequence, in different explanations needs. More precisely, it seems that these two kinds of people--end-users and database designers--do not necessarily "see" the same thing facing the same conceptual schema. In other words, the same conceptual schema does not *mean* the same thing for each of them. End-users figure out a tool which must be necessary for helping them in their activities. Database designers figure out the result of an information structuring activity, where many options were possible, and only some of them were chosen given a set of design goals.

These different *perspectives* on the same object, which include the agent's knowledge and concerns, may result in different *interpretations* of the same solution proposal. For instance, the sentence "For each employee, we need to store his name" would be understood:

(i) By the end-users, as a description of an information requirement;

(ii) By another database designer, not only as a description of an information requirement, but also as a structure of two kinds of data, one abstract datum, the entity "Employee," and one physically represented datum, the employee's name.

Given their perspective, the users reach a specific interpretation of the designer's solution proposal. We postulate that the interpretative process ends when the users find how this solution proposal is compatible with a response of agreement. In order to produce such a response, the users must check some dimensions of the solution proposal. We term these dimensions *acceptance goals.* An acceptance goal may be defined more precisely as a desired conclusion, imposed by the task, and that must be deduced from one solution proposal and some contextual knowledge.

Thus, in a validation task, end-users may want to know what such information, for instance the employee's name, is needed for. This is necessary for expressing an agreement with the solution proposal. The corresponding acceptance goal may be termed "X is useful." For the same proposal, the database designer may want to check if it will not be redundant with another datum, such as the employee's number. The corresponding acceptance goal may be termed "X is not redundant with Y".

Saying that the users' interpretative process ends when they find how the solution proposal is compatible with a response of agreement implies the following proposal: Users must *contextualize* the solution proposal in order to reach the acceptance goals. In this sense, the interpretative process is goal-driven.

Moreover, we may say that the need for an explanation arises if one of these acceptance goals cannot be reached. For instance, the end-user may ask "What is X useful for ?" if s/he is unable to contextualize X in order to interpret "X is useful". Or the database designer may ask "Why is Y not sufficient ?" if s/he is unable to contextualize X in order to interpret "X is not redundant with Y". Note that the same information given in the same task context provokes different acceptance goals, which may then induce different explanations needs (i.e., different questions).

This is equivalent to saying that an explanation is what bridges the gap between an interpretation that an agent is able to construct and the task-dependent conclusions s/he wants to reach (see Leake, 1991 for an account of the dependency between agent's goals and explanation needs.)

3.3.5 Conclusion

A difficulty for accepting an agent's proposal may lead the cooperation to improve not only the shared understanding of the problem to be solved, but also the agents' shared knowledge. These improvements are possible because of the explanations provided by both agents. Moreover, these improvements must lead to a better coordination between the agents, and consequently, a better cooperation. In this sense, we may consider *explanation as a means of the cooperation.*

Explanations could serve the cooperation only if both agents--the system and the user--can explain their proposals and critics. The ability to take users' explanations into account includes the need for (i) acquiring information from the user, (ii) propagating and testing the consequences of this information in the knowledge bases of the system, (iii) subsequently, asking for more information from the user, and (iv) eventually, acquiring new knowledge. Some of these capabilities are already present in some recent work aimed at designing cooperative systems, but under the user's control (e.,g., see Fischer, 1990; Edmonds & Ghazikhanian, 1991; Levrat & Thomas, 1993).

Other studies seem relevant to be considered in order to gain elements for implementing these requirements, particularly from the field of Knowledge Acquisition. Recently, some authors have proposed that this task should be considered as being accomplished in an incremental manner (e.g., see Karbach et al., 1990.) The system receives a loose specification as input and outputs a more comprehensive specification that provides the characterization of the domain. The overall goal is then to detail the structure and purpose of the domain in the context of its evolving theory (see Brézillon, 1994 for a review of the systems coupling knowledge acquisition and explanation.)

We have concluded that the way a cooperation progresses depends on explanations. We can also conclude that explanation needs depend on the context of the cooperation. Indeed, in a cooperation, agents try to reach a solution that must be mutually agreed. Then, any proposal must be accepted by the other agents. This acceptance phase relies on the explanatory power of the information that is provided. The observations reported above shows that this feature is dependent on contextual factors such as the task at hand, and other agents' knowledge.

We assume that the modelling of perspectives proposed in McKeown et al. (1985) could represent an interesting solution for implementing these ideas. The authors propose that perspectives should be represented with intersecting multiple hierarchies. The hierarchies are cross-linked by entities and processes which can be viewed from different perspectives. This kind of partitioning of the knowledge base allows the generation system to distinguish between different types of information that support the same fact.

We now turn to the study of the explanation process itself, without consideration of the task context.

3.4 Cooperation for explaining

Once an explanation need arises in a dialogue, how is it satisfied? The data collected from cooperative dialogues show that both agents--the explainer and the explainee--attempt to reach the explanatory goal *in a joint effort.* We use the terms "explainer" and "explainee" for describing two roles that can be taken by either agents during a process of explanation. Inside a given exchange, each role can be speaker or hearer (for convenience, we will use "he" for referring to the explainer, and "she" for referring to the explainee). Thus, we note that:

(i) Explanation needs are not necessarily revealed by a question; Most of the time they are predicted, and explanations are produced spontaneously;

(ii) When a request for an explanation appears in a dialogue, contextual information helping the explainer in answering the request may also be conveyed;

(iii) Explainee's interventions may complete the explainer's first explanatory attempt.

Some previous studies have already discussed the collaborative nature of the explanation process [e.g., Gilbert, 1989; Moore & Swartout, 1990; Cawsey, 1993]. However, we believe that the collaboration depicted by these studies does not reflect the richness of the possibilities that two agents can use for achieving an explanation goal. Our motivation in trying to reflect all these possibilities lies on two facts: First, to our knowledge, they are not described in the literature on explanation; Second, we believe that the more advanced systems built up until now are still unable to solve some misunderstandings, especially when the gap between the system's and the user's knowledge is large. One of the main reasons is that most of the user's collaborative power is not taken into account.

3.4.1 Spontaneous explanations

Most of explanations are given spontaneously in naturally occurring dialogues. Thus, in validation dialogues, we pick out three reactive explanations against 25 spontaneous explanations of proposed solution on average per dialogue (83% of spontaneous explanations).

Other researchers also note this surprising significance of spontaneous explanations. Cahour and Darses (1993) study diagnosis dialogues recorded in a sewage plant between an operator and an engineer. They found that eighty per cent of the explanations collected were volunteered. Belkin (1988) has also pointed out that the help given by an adviser during information-seeking dialogues is also characterized by the superiority of spontaneous explanations on reactive explanations.

An argument that may account for such a phenomenon is the fact that agents are *mutually dependent*. Each piece information that is provided by one agent to another is useful for the receiver and indirectly for the provider. Indeed, the actions and results of the other will depend on the information that is provided and understood. Moreover, the provider's goals rely on the other's results. Thus, one may understand why the provider wants to be understood and why he produces so many spontaneous explanations.

Spontaneous explanations appear in three ways (see Draper, 1987):

(i) When submitting new or conflicting information,
(ii) When detecting a misconception from the other agent's questions or assertions [McCoy, 1988],
(iii) When detecting an obstacle in the other agent's plan (e.g., Allen & Perrault, 1980).

We will only consider the first case here.

We do not find spontaneous explanations with any proposed solution. In database validation dialogues, twenty-nine per cent of the solution proposals (in total, 64 from 222) are spontaneously explained. Moreover, we notice that these explanations are aimed at three different functions:

- *Fill in missing knowledge required to check task-dependent acceptance goals*
  Example: "We need the 'order number' because it will allow you to retrieve all the articles ordered."
  In this sentence, the designer points out the reason why the data 'order number' is useful. We consider "X is useful" as an acceptance goal that is dependent on the validation task. Most of the explanations belonging to this class were provided with the proposal of *new* elements of the conceptual schema[5], or when a procedure of use of some data was *unusual* (for instance, because one datum has not to be entered by the user, but is produced by the machine).
- *Prevent or resolve a conflict between some users' expectations and the solution proposal.*
  Example: "For each garment, we are going to store the 'size'. Well, at the beginning, I didn't write this because the machine must determine the size, given the employee's name and the garment ordered. But there are a few cases where we need to manually enter this information [describes the cases]."
  The previous precedent solution acts here as an users' expectation, which is contradicted by the designer's new proposal. The explanation modifies the context in which the new proposal has to be considered.
  This kind of explanation may be given before or after presenting a proposed solution. In the first case, the explainer tries to prevent a conflict between what was expected and what is proposed, while in the second case, the explainer attempts to resolve this kind of conflict.
- *Argue for a proposal P being insufficiently founded.*
  Example: "We will not store the value of each garment, because it seems that it is constantly changing. (Indeed) X said to me that the suppliers changed their price almost every week because …."
  The second part of the explanation ("X said to me …") aims at supporting the belief that "The value of the garments is constantly changing." Such an explanation goal is very similar to the first one. The designer provides the users with missing knowledge that is needed to check an acceptance goal. However, the acceptance goal that is relevant here is not directly linked to the task. Instead, it is related to the information conveyed in the explanation.

These three types of spontaneous explanation point out that there are three situations in which an acceptance goal cannot be reached: Either users miss contextual knowledge which is necessary to deduce the acceptance goals from an interpretation of a solution proposal; the users' context contain false assumptions, which could lead them to reject a given proposed

[5] We were able to check the novelty of the proposed solutions because (i) we were present in some of the previous meetings between the designer and the end-users, and consequently were able to say if a given proposal had been already discussed, (ii) reports of previous meetings with the end-users were available, (iii) the designer might present a given proposal as being new : "I discover yesterday this need. So we need to store these data"

solution; users miss contextual knowledge that is necessary to accept the information included in an explanation.

Karsenty and Falzon (1992) interpret spontaneous explanations as an anticipation of the users' explanation needs. This anticipation ability must rely on:

(i) a model of users' knowledge and expectations (the users' context); This gives the possibility of reaching an interpretation close to that of the users' one.

(ii) a model of the users' acceptance goals related to a given proposed solution in a given task.

The significance of spontaneous explanations seems to reveal that the designer *must* bridge the gap between the users' context and his own context each time that the discrepancies between them could result in a users' disagreement. Moreover, we may assume that not providing spontaneous explanations is equivalent to *communicating all relevant information that the users can infer* [6] for validating a solution proposal (the meaning of the data proposed, their necessity, their procedure of use, and so on).

A direct consequence of this assumption is that *explicit* (i.e., not predicted) requests can be seen as cases when the designer has an inaccurate model of other agents' context [Karsenty & Falzon, 1992]. This suggest that the explainer's processing of unexpected questions (particularly, follow-up questions) must include a revision of his model of the other agents' context.

This analysis stresses the significance of the notion of context, which is necessary (i) for interpreting what a speaker means when s/he produces a message, (ii) for anticipating what hearer will believe that a speaker means. The significance of the notion of context is well-recognized in various domains like Natural Language and Human-Machine Communication (e.g., see Cahour & Karsenty, 1993; Careni et Moore, 1993; Mittal & Paris, 1993.) However, there are few implemented systems that explicitly deal with context. A reason is that few formalisms of knowledge representation permit an effective use of context (some first attempts may be found in McCarthy, 1993; Sowa, 1992.)

3.4.2 "Contextualization" of explanation questions

Traditionally, we consider a question as a proposition expressed under an interrogative form. For instance, "Why do you need this information?," "How did you reach this conclusion?"

In real dialogues, the explainee may provide the explainer with contextual information with his question. We distinguish two cases that seem particularly relevant to the handling of explanation needs:

1. Question and answer assumption

Example: "Is this built with two parts because of some specific problems of assembly?"
In this utterance, we may distinguish the question "Why is this built with two parts?" and an answer assumption "because of specific problems of assembly." This kind of contextualization has the advantages of:

(i) Reducing the explainer's effort and allowing a point of mutual understanding to be reached more rapidly if the answer confirms the explainee's assumption;

(ii) Allowing the explainer to predict an explanation need if the answer differs from the explainee's assumption.

2. Question and question rationale

Example: "You said that we will ask the supplier for recording a further paper for each order. I know by experience that such papers get lost, what creates new problems, I let you figure it out. So, isn't there a more realistic solution?"

Here, the question is the last sentence of the intervention. (Note the difference if just this sentence had been uttered.) The two first sentences constitute the contextual information that describes the reasons for the request.

Such a contextualization of the question may take the form of a linguistic mark referring to the previous part of the dialogue where the question rationale lies.
Example (A and B are two persons, and numbers organize utterances):

A1 Is this part a pivot?
B1 No, this is an embedding.

[6] This is a direct consequence of the Gricean principle of cooperation and the maxims of conversation [Grice, 1975].

A2 *Then,* what is the function of this axle?

In this example, A utters A2 because of B's negative answer to A1. The marker "then" refers to the sequence A1-B1.

We believe that this kind of contextualization aims at guiding the explainer in finding an appropriate explanation. We base our assumption on the model of explanation that arises from some psychological studies.

For instance, Hilton says that "To understand a request for an explanation, we must know the implicit contrast that it presupposes, the rest is logic" [Hilton, 1988, p. 58]. Another quotation may complete this one [Turnbull & Slugoski, 1988, p.68]: "A felicitous answer consists of the provision of information unknown to the asker that can resolve the asker's puzzle. Including their state of mutual knowledge, the knowledge states of asker and askee, are central to the explanation process."

Then, with question rationale, the explainer can better comprehend--and better resolve--the nature of the cognitive conflict that provokes the explanation need when the explainee informs the explainer of her question rationale.

We believe that this kind of contextualization reflects the same principle that explains the significance of spontaneous explanations. Askers *must* bridge the gap between the askee's context and their own context each time that the discrepancies between them could result in a difficulty in answering or in the providing of an unsuitable answer.

The quantitative significance of spontaneous explanations leads us to believe that a speaker who initiates a proposal, wants to help the hearers in understanding it. With the examples of contextualization of a question presented above, we notice that the hearers may also attempt to help the speaker to find the appropriate answer.

3.4.3 The process of explanation

Often, an act of explaining is not a single intervention. There is a need for a mutual agreement through a mixed-initiative dialogue where both the explainer and the explainee act together. Such a co-participation may take different forms:

- An *explainee regulation of the explanatory information flow*
  One obtains such a regulation with linguistic marks of agreement (e.g., 'OK', 'mmh', etc.) that cut out the explanation. It is important to note that very often these agreement marks are sought by the explainer. This means that he must decide when to cut an explanation to control the explainer assimilation process. Further studies are needed to understand more precisely how this is done.
- *Follow-up questions*
- *Explainee's context description*
  Such a description is used when the hearer fails to understand the speaker. It consists of describing the reasons that prevent her from understanding the speaker's utterance. This is particularly clear when the answer to a follow-up question does not satisfy the explainee. The following example, taken from a real phone conversation, illustrates this possibility [Cahour & Karsenty, 1993]:

  A and B know each other. A is an ex-student and B a professor.
  A1 I call you to see if you could write for me a letter of recommendation for an academic situation.
  B1 Oh... yes... oh... but... I don't quite understand... are we talking of the same thing?
  B2 Mr. Y called me and told me that you were interested by requiring this situation in our laboratory.
  A2 Oh, I didn't know,
  A3 And since then I happened to know that the possible laboratories are defined beforehand and your laboratory is not in the list.

  B makes his context explicit with the utterance B2. Note that in doing so, he explains his unexpected question ("Are we talking of the same thing ?"). This allows A to detect more easily a mistake in B's representation of A's intent, and to correct it (A2 and A3). This ability gives human agents the power for repairing most of the communication failures.
- *Rephrasing of an explanation by the explainee*
  This also allows agents to reach a point of mutual understanding. The explainee rephrases the explanation, and asks the explainer for a confirmation at the same time.
  Example:
  A1 Otherwise, maybe we need to index them according to the angle […] in order to set this stuff upon that one.
  B1 Ah OK, *because you have an indexing between this and that also* [rephrasing]

A2 Yes, that's it.
B2 OK, OK.

Actually, it is better to say that the explainee makes his context explicit *once the explanation was accepted*. Rephrasing the explanation with her words, the explainee just expresses the changes that it provokes in her mind. The explainer's confirmation ensures the explainee that she has understood the speaker's utterance.

4 CONCLUSION

In the expert system framework, one agent--the system--achieves the problem solving process. In a cooperative problem solving process, two agents at least may develop a reasoning and achieve parts of the common task. This dictates most of the explanation features in the cooperation framework. We give below some of these features:
(1) Bilateral explanations,
(2) The explanation process as part of problem solving,
(3) Relation between problem solving and knowledge acquisition mediated by the explanation process,
(4) An explanation process driven by acceptance goals related to the task at hand,
(5) Significance of spontaneous explanations, caused by the mutual dependency between agents.

The last feature seems to be more general, and less dependent on a cooperative work setting:
(6) The explainer and the explainee must cooperate to achieve an explanation goal.

On the one hand, the explainer anticipates the explainee's needs and looks actively for cues indicating how his explanation is understood. On the other hand, the explainee may act in many ways for making the speaker's talk more understandable. Moreover, we have proposed a view of the communication process where speakers only volunteer explanations when they can predict that the hearer's context is not appropriate for inferring the full intended meaning of a proposal. As a consequence, uttering a proposal without any spontaneous explanation is equivalent to communicating all the relevant inferences that the other agent can draw.

This study of human-human cooperative dialogues, by describing the complexity of the explanation process, leads to the following conclusion. It is still difficult to believe that a self-explainable system can produce satisfying explanations, especially when the gap between agents' knowledge (the user and the system) is large. In particular, this would require natural-language processing capabilities, and models for managing the communication with the user that are still not available (see Baker, 1992 for some interesting insights on handling the problem of complex dialogue management.) However, we hope that our analysis of the explanation features in a cooperation will help in building a better computational model for the design of cooperative systems.

REFERENCES

Allen J.F. & Perrault C.R. (1980) Analyzing Intention in Utterances. *Artificial Intelligence*, 15, 143-178.

Alvarez I. (1992) Morphological explanation: a strategy based on the study of the case to explain. *Proc. of the ECAI Workshop on "Improving the Use of Knowledge-Based Systems with Explanations",* Vienna, Austria. Research Report 92/21, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75005 Paris cedex 05, France. 57-64.

Baker M. (1992) Analysing rhetorical relations in collaborative problem solving dialogues. *Proc. of NATO Workshop "Natural Dialogue and Interactive Student Modelling"*, Varenna, Italy, Oct. 16-19.

Baker M., Dessalles J.L., Joab J.L., Raccah P.Y., Safar B. & Schlienger D. (1993) Analysis and modelling of negotiated explanations. *Proc. of the Workshop on Explanation and Cooperation*, Research Report N° 105, CNAM, Laboratoire d'Ergonomie, 41 rue Gay-Lussac, 75005 Paris, France (in French).

Belkin N.J. (1988) On the nature and function of explanation in intelligent information retrieval *Proc. of the 11th Conf. on Research and Development in Information Retrieval,* Grenoble, France. 135-145.

Brézillon P. (1992) Architectural and contextual factors in explanation construction. *In Proc. of the IJCAI Workshop on "Improving the Use of Knowledge-Based Systems with Explanations",* Vienna, Austria. Research Report 92/21, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75005 Paris cedex 05, France. 65-74.

Brézillon P. (1994) Design of an Intelligent Assistant System from Several Application, *Proc. of the International Conference for Expert Systems for Development*, Bangkok, March (to appear).

Cahour B. & Darses F. (1993) *Cognitive Processing and Explanation Strategies in a Diagnosis Task.* ISEE Deliverable D140.2, Esprit project 6013.

Cahour B. & Karsenty L. (1993) Context of Dialogue: A Cognitive Point of View. *Proc. of the IJCAI'93 Workshop on Using Knowledge in its Context.* Chambéry, France, August 29. Research report 93/13, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75005 Paris cedex 05, France. 20-29.

Carenini G. & Moore J.D. (1993) Generating Explanation in Context. *Proc. of the International Worshop on Intelligent User Interface*. Orlando.

Carr C. (1992) Performance Support Systems: A New Horizon for Expert Systems. *AI Expert*, May 1992. 44-49.

Cawsey A. (1993) Planning Interactive Explanations. *International Journal of Man-Machine Studies*, 38(2), 169-200.

Cawsey A., Galliers J., Reece S. & Sparck Jones K. (1992) The Role of Explanation in Collaborative Problem Solving. *Proc. of the ECAI-92 Workshop "Improving the Use of Knowledge-Based Systems with Explanations*". Vienna, Austria.

Chandrasekaran B., Tanner M.C. & Josephson J.R. (1989) Explaining control strategies in problem solving. *IEEE Expert*, Fall 1989, 9-24.

Clancey W.J. (1983) The Epistemology of a Rule-Based Expert System - A Framework for Explanation. *Artificial Intelligence*, 20, 215-251.

Coombs M. & Alty J. (1984). Expert systems: an alternative paradigm. *International Journal of Man-Machine Studies*, 20, 21-43.

Darses F., Falzon P. & Robert J.M. (1993) Cooperating partners: investigating natural assistance. In: Salvendy G. & Smith M.J. (Eds.) *Human-Computer Interaction: Software and Hardware Interfaces.* Elsevier.

Dieng R., Giboin A., Tourtier P.A. & Corby O. (1992) Knowledge-acquisition for explainable, multi-expert knowledge-based design systems. In: Wetter T., Althoff K.A. Boose J. Gaines B. Linster M. & Schmalhofer F. (Eds.)*Current Developments in Knowledge Acquisition: EKAW-92* . Springer-Verlag, Berlin.

Draper S.W. (1987) The Occasions for Explanation. *Proc. of the 3rd Alvey Explanation Workshop*, Guilford, September 1987, 199-207.

Edmonds E. & Ghazikhanian J. (1991) Cooperation Between Distributed Knowledge-Bases and the User. In: Weir G.R.S. & Alty J.L. (Eds.) *Human-Computer Interaction and Complex Systems.* London: Academic Press.

Falzon P. (1991) Cooperative Dialogues. In: J. Rasmussen, B. Brehmer & Leplat J. (Eds.) *Distributed Decision Making: Cognitive Models for Cooperative Work.* Chichester, UK: Wiley.

Fischer G. (1990) Communication Requirements for Cooperative Problem Solving Systems. *Information Systems*, 15(1), 21-36.

Gilbert N. (1987) Question and Answer Types. In: Moralee D.S. (Ed.) *Research and Development in Expert Systems III.* Cambridge University Press.

Gilbert N. (1989) Explanation and Dialogue. *The Knowledge Engineering Review*, 4(3), 235-247.

Grice H.P. (1975) Logic and Conversation. In: Cole P. & Morgan J.L. *Syntax and Semantics*, vol.3. NY: Academic Press.

Hasling D.W., Clancey W.J. & Rennels G. (1984) Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies*, 20, 3-19.

Hilton D.J. (1988) Logic and Causal Attribution. In: Hilton D.J. (Ed.)*Contemporary Science and Natural Explanation. Commonsense Conceptions of Causality.* Harvester Press.

Hughes S. (1986) Question Classification in Rule-Based Systems. In: Brauer M.A. (Ed.) *Research and Development in Expert Systems III.* Cambridge University Press.

Jefferson G. (1972) Side sequences. In: Sudnow D.N. (Ed.) *Studies in social interaction.* New York: Free Press.

Karbach W., Linster M. & Voss A. (1990) Models, methods, roles and tasks: many labels - one idea ?*Knowledge Acquisition*, 2, 279-299

Karsenty L. & Falzon P. (1992) Spontaneous explanations in cooperative validation dialogues. *Proc. of the ECAI-92 Workshop on Improving the use of knowledge-based systems with explanations*. Vienna, Austria. Research Report 92/21, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75005 Paris cedex 05, France.

Keravnou E.T. & Washbrook J. (1989) What is a deep expert system ? An analysis of the architectural requirements of second-generation expert systems. *The Knowledge Engineering Review*, 4(3), 205-233.

Kidd A.L. (1985) What Do Users Ask ? Some Thoughts on Diagnostic Advice. In: Merry M. (Ed.) *Expert Systems 85.* Cambridge University Press.

Leake D.B. (1991) Goal-Based Explanation Evaluation. *Cognitive Science*, 15, 509-545.

Lemaire B. & Safar B. (1991) Some necessary features for explanation planning architectures: study and proposal *Proc. of the AAAI'91 Workshop on Explanation*, Anaheim, USA, July 1991.

Levrat B. & Thomas I. (1993) Tailoring Explanations to the User's Expectations: A Way to Be Relevant. *Proc. of the IJCAI'93 Workshop on Explanation and Problem Solving*. Chambéry, France, August 29.

Maïs C. & Giboin A. (1989) Helping users achieve satisficing goals. In: Smith M.J. & Salvendy G. (Eds.) *Work with Computers: Organizational, Management, Stress and Health Aspects*. Elsevier Science Publishers: Amsterdam. 98-105.

Malhotra A., Thomas J.C., Carroll J.M. & Miller L.A. (1980) Cognitive processes in design. *International Journal of Man-Machine Studies*, 12, 119-140.

McCarthy J. (1993) Notes on formalizing context. *Proc. of the 13th IJCAI-93*, Chambéry, France. 555-560.

McCoy K.F. (1988) Reasoning on a Dynamically Hilghlighted User Model to Respond to Misconceptions. *Computational Linguistics*, 14(3).

McKeown K.R., Wish M. & Matthews K. (1985) Tailoring Explanations for the User. *Proc. of IJCAI'85*, 794-798.

Mittal V. & Paris C.L. (1993) Context: Identifying its elements from the communication point of view. *Proc. of the IJCAI'93 Workshop on Using Knowledge in its Context*. Chambéry, France, August 29. Research report 93/13, LAFORIA, Box 169, University Paris 6, 4 place Jussieu, 75005 Paris cedex 05, France. 87-97.

Moore J.D. & Swartout W.R. (1990) A Reactive Approach to Explanation: Taking the User's Feedback into Account. In: Paris C.L., Swartout W.R. & Mann W.C. (Eds.) *Natural Language Processing in Artificial Intelligence and Computational Linguistics*. Kluwer Academic Publishers.

Nisbett R.E. & Ross L. (1980) *Human inferences: Strategies ans shortcomings of social judgement*. Englewood Cliffs, NJ: Appleton-Century- Crofts.

Paris C.L. (1990). Generation and Explanation: Building an Explanation Facility for the Explainable Expert Systems Framework. In: Paris, Swartout, Mann (Eds.) *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic Publishers.

Paris C.P., Wick M.R. & Thompson W.B. (1988) The Line of Reasoning Versus The Line of Explanation. *Proc. of the AAAI'88 Workshop on Explanation*. Anaheim, USA, July 1991.

Perrot L., Brézillon P., Fauquembergue P. (1993) Towards automatic generation of knowledge bases for diagnosis systems in the field of power systems. *Proc. of ISAP'93*, January 1993.

Pollack M.E. (1985) Information Sought and Information Provided: An Empirical Study of User/Expert Dialogues. *Proc. of CHI'85* San Francisco. 155-159.

Pollack M.E., Hirschberg J. & Webber B. (1982) User Participation In The Reasoning Processes of Expert Systems. *Proc. of AAAI-82, National Conference on Artificial Intelligence*. 358-361.

Sowa J.F. (1992) Representing and reasoning about contexts. *Proc. of the AAAI'92 Workshop on Propositional Knowledge Representation*, Stanford, CA. 133-142.

Suchman L. (1987) *Plans and situated actions: The problem of human-machine communication.* Cambridge, UK: Cambridge University Press.

Swartout W.R. (1983) XPLAIN: a System for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence*, 21(3), 285-325.

Teach R.L. & Shortliffe E.H. (1984) An Analysis of Physicians' Attitudes. In: Buchanan B.G. & Shortliffe E.H (Eds.) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, Mass: Addison-Wesley.

Turnbull W. & Slugoski B.R. (1988) Conversational and Linguistic Processes in Causal Attribution. In: Hilton D.J. (Ed.) *Contemporary Science and Natural Explanation. Commonsense Conceptions of Causality.* Harvester Press.

Van Beek P. (1987) A Model For Generating Better Explanations. *Proc. of the 25th Conference of the Association for Computational Linguistics*, Stanford, CA. 215-220.

Wallis J.W. & Shortliffe E.H. (1984) Customized Explanations Using Causal Knowledge. In: Buchanan B.G. & Shortliffe E.H. (Eds.) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, Mass: Addison-Wesley.

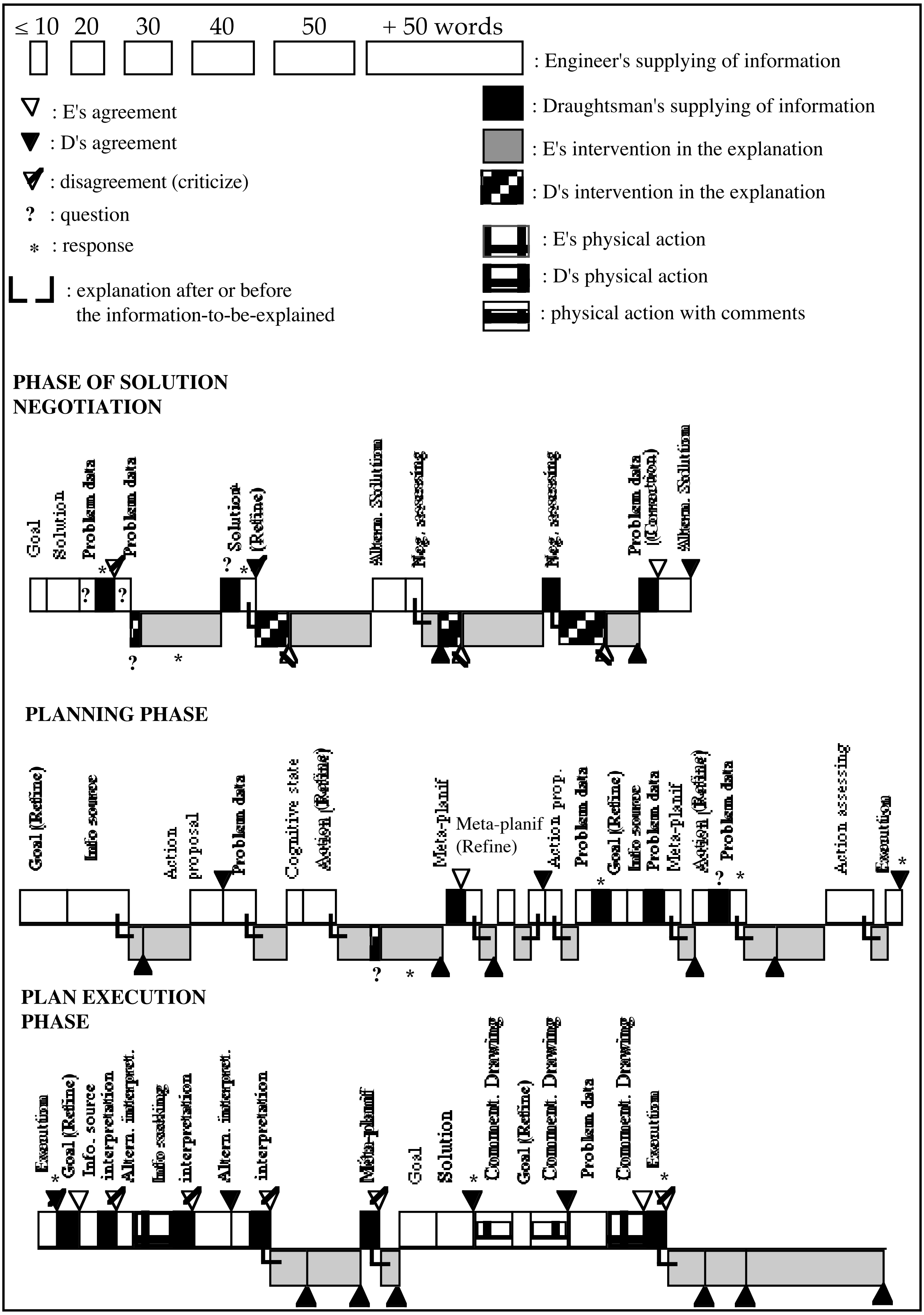
Wick M.R. & Thompson W.B. (1989) Reconstructive Explanation: Explanation as Complex Problem Solving. *Proc. of the 11th International Joint Conference on Artificial Intelligence*, Detroit, Michigan, August 20-25.

Woods D.D & Hollnagel E. (1987) Mapping cognitive demands in complex problem solving worlds. *International Journal of Man-Machine Studies*, 26, 257-275.

Woods D.D. & Roth E.M. (1988) Cognitive Systems Engineering. In: Helander M. (Eds.) *Handbook of Human-Computer Interaction.* North-Holland: Elsevier.

Woods D.D., Roth E.M. & Benett K. (1990) Exploration in joint human-machine cognitive system. In: Robertson S., Zachary W. & Black J.B. (Eds.) *Cognition, Computing and Cooperation* . Norwood, NJ: Ablex.

Figure 1: Graphical examples of cooperative design dialogues



Int. J. Expert Systems with Applications,, 8(4): 445-462 (1995)