Explaining for Contextualizing and Contextualizing for Explaining

Patrick BREZILLON

LIP6, case 169, University Paris 6
104 avenue du President Kennedy, 75016 Paris, France
Patrick.Brezillon@lip6.fr

Abstract. This paper proposes a view on the relationships between explanation and context. First, we install the background of our proposal. This background comprises two parts: the consideration of explanations in knowledge-based systems, and a preliminary observation of relationships between explanations and context. We comment briefly previous works on explanations in order to point out what is reusable. Second, we discuss a set of new types of explanation in a context-based formalism called contextual graphs. We begin by presenting the context-based formalism of representation and after an explanation typology that can be established, thanks to Contextual Graphs.

1 Introduction

Any collaboration supposes that each participant understands how others make a decision and follows the series of steps of their reasoning to reach the decision. In a face-to-face collaboration, participants use a large part of contextual information to translate, interpret and understand others' utterances by using contextual cues like mimics, voice modulation, movement of a hand, etc. Explanation generation relies heavily on contextual cues [10] and thus would play now a role in e-collaboration more important than in face-to-face collaboration.

Twenty years ago, Artificial Intelligence was considered as the science of explanation [12]. However, there are few concrete results to reuse now from that time (e.g. see [15]). There are several reasons for that. The first point concerns expert systems (and knowledge-based systems after) themselves and their past failures [9]:

- There was an exclusion of the human expert providing the knowledge for feeding the expert systems. The "interface" was the knowledge engineer asking the expert "If you face this problem, which solution do you propose?" The expert generally answered something like "Well, in the context A, I will use this solution," but the knowledge engineer retained the pair {problem, solution} and forgot the initial triple {problem, context, solution} provided by the expert. The reason was to generalize in order to cover a large class of similar problems when the expert was giving a local solution in a specific context. Now, we know that a system needs to acquire knowledge and its context of use.
- On the opposite side, the user was excluded from the noble part of the problem solving because all the expert knowledge was supposed to be in the machine: the machine was considered as the oracle and the user as a novice [10]. Thus,

explanations aimed to convince the user of the rationale used by the machine without respect to what the user knew or wanted to know. Now, we know that we need of a user-centered approach [5].

- Capturing the knowledge from the expert, it was supposed to put all the needed knowledge in the machine, prior to the use of the system. However, one knows that the exception is rather the norm in expert diagnosis. Thus, the system was able to solve 80% of the most common problems, on which users did not need explanations and nothing about the 20% that users did not understand. Now, we know that systems must be able to acquire incrementally knowledge with its context of use in order to address more specific problems.
- Systems were unable to generate relevant explanations because they did not consider what the user's question was really, in which context the question was asked. The request for an explanation was analyzed on the basis of the available information to the system. Now, we know that the system must understand the user's question and after build jointly with the user the answer.

Thus, the three key lessons learned are: (1) KM (i.e. knowledge management normally) stands for management of the knowledge in its context; (2) any collaboration needs a user-centered approach; and (3) an intelligent system must incrementally acquire new knowledge and learns corresponding new practices. We present in [3] and [7] a context-based formalism for explaining concretely the differences often cited but never clearly identified between prescribed and effective tasks [13], procedures and practices [4], logic of functioning and logic of use [16]).

Focusing on explanation generation, it appears that a context-based formalism for representing knowledge and reasoning allows to introduce the end-user in the loop of the system development and to generate new types of explanations. Moreover, such a formalism allows a uniform representation of elements of knowledge, of reasoning and of contexts.

Hereafter, the paper is organized in the following way. First, we install the background of our proposal. This background comprises two parts: the consideration of explanations in knowledge-based systems, and the relationships between explanations and context. We comment briefly previous works on explanations in order to point out what is reusable. Second, we discuss a set of new types of explanation in a context-based formalism called contextual graphs. We begin by presenting the context-based formalism of representation and continue after by a presentation of an explanation typology that can be established, thanks to Contextual Graphs.

2. Background

This section introduces briefly the evolution of the way in which explanations have been considered in experts systems and after in knowledge-based systems. In a second part, we show that it was clear that there is a relationships between explanation generation and context, the lack of concrete works on context at that time (end of 80's) has seriously limited the interest of explanations in knowledge systems.

2.1 Explanations in Expert Systems and Knowledge-Based Systems

The first research on explanations started with rule-based expert systems. Imitating a human reasoning, the presentation of the trace of the expert system's reasoning (i.e. the sequence of fired rules) was supposed to be an explanation of the way in which the expert system reaches a conclusion. Indeed, it was right, but explanations were generated at the implementation level. The following step was the use of canned texts where "Firing of Rule_23 allows to checking rule_7" was replaced by something like "The available facts allow to identify the failure on equipment piece B3, and this leads to check if it is a mechanical problem". Explanations thus moved from the implementation level to a representation level. However, the logic behind the chaining of the rules (why rule_7 is chosen first for example) was hidden. An important reason discovered lately is that a part of the control knowledge was put in the inference engine implicitly by the knowledge engineer (by imposing the ordering of rule checking for example). Thus, it was not possible to go another step above (i.e. a modeling level after the implementation and representation levels).

Rapidly, it was clear that it was not possible to explain heuristics provided by human experts without additional knowledge. It was then proposed to introduce a domain model. It was the second generation of expert systems, called the knowledge-based systems. This approach reached also its limits because it was difficult to know in advance all the needed knowledge and also because it was not always possible to have models of the domain. The user's role was limited to be a data gatherer for the system. A second observation was that the goal of explanations is not to make identical user's reasoning and the system reasoning, but only to make them compatible: the user must understand the system reasoning in terms of his own mental representation. For example, a driver and a garage mechanic can reason differently and reach the same diagnosis on the state of the car. The situation is similar in collaboration where specialists of different domains and different geographical areas must interact in order to design a complex object. A third observation is that the relevance of explanation generation depends essentially on the context use of the topic to explain [10], [1].

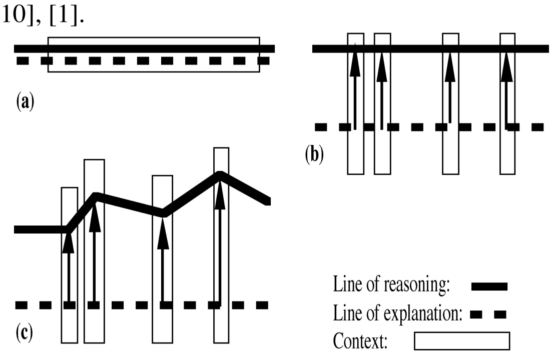


Fig. 1. Line of reasoning versus line of explanation [1]

Even if expert systems are now abandoned, there are important results that we can yet reuse, such as the base for new explanations proposed by Spieker [19] and the qualities for relevant explanations established by Swartout and Moore [20]. Thus, beyond the need to make context explicit, first in the reasoning to explain, and, second, in the explanation generation, the most challenging finding is that lines of reasoning and explanation must be distinguished. Figure 1 illustrates the evolution of the research on explanation generation [1]. Figure 1.a gives the initial view on explanation generation by a strict superposition of the lines of reasoning and explanation (the firing of rule 23 allows to check Rule7). Figure 1.b represents the first evolution corresponding to the introduction of domain knowledge, the knowledge that is not necessary for reasoning but for explanation. This was the first separation of the line of reasoning and the line of explanation. Figure 1.c shows that lines of reasoning and of explanation interact, and providing an explanation may modify the line of reasoning. Thus, the line of explanation was considered during the development of the line of reasoning and not produced after the reasoning of the system.

Thus the key problem for providing relevant explanations is to find a uniform representation of elements of reasoning and of context.

2.2 Explanations and Contexts

A frequent confusion between representation and modeling of the knowledge and reasoning implies that explanations are provided in a given representation formalism, and their relevance depend on explanation expressiveness through this formalism. For example, ordinary linear differential equation formalism will never allow to express—and thus explaining—the self-oscillating behavior of a nonlinear system. Thus, the choice of representation formalism is a key factor for generating relevant explanations for the user and is of paramount importance in collaboration with different users and several tasks.

A second condition is to account for, make explicit, and model the context in which knowledge can be used and reasoning held. This concerns the needed distinction between data, information and knowledge. For example, a temperature of 24°C (the datum) in winter in Paris (when temperature is normally around 0°C) is considered to be hot (the "French information") and cold (the "Brazilian information") in Rio de Janeiro (when temperature is rather around 35°C during winter). Thus, the knowledge must be considered within its context of use for providing relevant explanations, like to explain to a person living in Paris why a temperature of 24°C could be considered as cold in some other countries. Temperature = 24°C is a datum. A process of interpretation leads to an information (hot or cold). Information is data with meaning built on the basis of the knowledge that the person possesses. The knowledge is specific to a person and constitutes the context in which a person evaluates (and eventually integrates) information pieces in his mental representation. Indeed, this is more particularly the part of the knowledge that the person finds more or less related to the information. It corresponds to a mental representation that the person built from its experience for giving meaning to the information and eventually integrates the information in the body of contextual knowledge already available. It is when a needed information cannot be related totally to the mental representation that an explanation is required for making explicit the links between the information and the contextual knowledge of the person. We will come back on this point on the following.

There is now a consensus around the following definition "context is what constrains reasoning without intervening in it explicitly" [8], which applies also in e-collaboration (although with more complex constraints) where reasoning is developed collectively. Explanation generation is a means to develop a shared context among the actors in order to have a better understanding of the others (and their own reasoning), to reduce needs for communication and to speed up interaction.

From our previous works on context, several conclusions have been reached. First, a context is always relative to something that we call the (current) focus of attention of the actors. Second, with respect to this focus, context is composed of external knowledge and contextual knowledge. The former has nothing to see with the current focus (but could be mobilized later, once the focus moves), when the former can be more or less related directly to the focus (at least by some actors). Third, actors address the current focus by extracting a subset of contextual elements, assembling and structuring them all together in a proceduralized context, which is a kind of « chunk of contextual knowledge » (in the spirit of the "chunk of knowledge" of Schank [17]). Fourth, the focus evolving, the status of the knowledge (external, contextual, into the proceduralized context) evolves too. Thus, there is a dynamics of context that plays an important role in the quality of explanations.

As the context exists with the knowledge, a context-based generation of explanations does not require an additional effort because the explanatory knowledge is integrated in the knowledge representation at the time of their acquisition and the representation of the reasoning (see [4] on this aspect). However, this supposes to have a context-based formalism allowing a uniform way to represent elements of reasoning and of contexts.

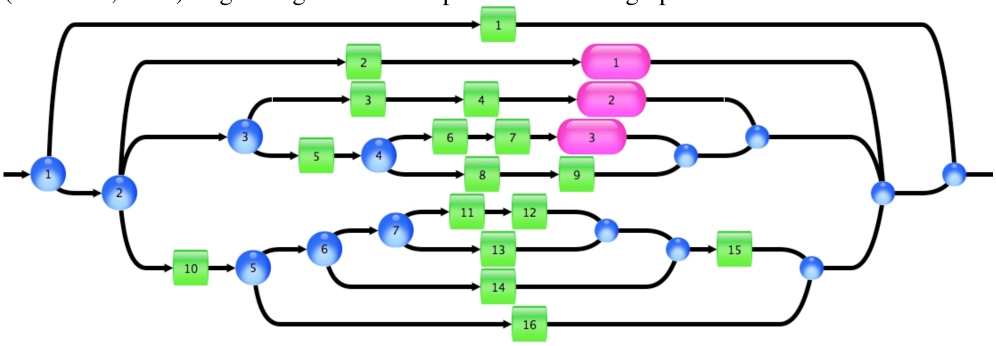
3. Contextual Graphs and Explanation Typology

In Section 2 we show that it was necessary to develop a conceptual framework for context modeling. In this section, we show now, first that the development of our conceptual framework lead to the implementation of Contextual Graphs, which allows a uniform representation of elements of reasoning and of contexts. Then, in such a representation formalism, we come back on the types of explanation that are possible to generated in contextual graphs because "explanatory knowledge" is a natural part of the knowledge in knowledge systems.

A key point here is that contextual graphs is a representation formalism as workflows, Petri nets, Bayesian nets, etc. However, the main difference is that Contextual Graphs is a user-centered formalism [5]: any user (e.g. a psychologist) needs less than one minute to learn and use the software (freely available at http://www.cxg.fr).

3.1 The Context-Based Formalism of Representation

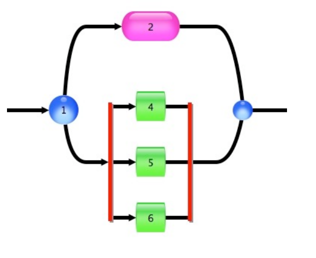
A contextual graph represents the different ways to solve a problem. It is a directed graph, acyclic with one input and one output and a general structure of spindle (Brezillon, 2005). Figure 2 gives an example of contextual graph.



1: Is the site already known?
Yes 1: Look for new stuff.
No 2: What is the link target?
Html page 2: Open the target in a new window.
1: Activity-1
PDF, doc or ps page
3: Is there an html version?
Yes 3: Open the target in the new window.
4: Look for keywords on a colored background.
2: Activity-1
No 5: Download the document.
4: Have I time now?
Yes 6: Open the document.
7: Look for keywords.
3: Activity-1
no 8: Store the document in "To read" folder.
9: Close the window.
ppt page 10: Open the target in a new window.
5: Duration of the download?
Short 6: Is it for a course?
Yes 7: Can page content be retrieved?
Yes 11: Copy the slide.
12: Paste slide in my document.
No 13: Note idea for later.
No 14: Look for other stuff.
15: Go to the next slide.
Long 16: Close the window.

Fig. 2. "Information retrieval" (from [4])

A path in a contextual graph corresponds to a specific way (i.e. a practice) for the problem solving represented by the contextual graph. It is composed of elements of reasoning and of contexts, the latter being instantiated on the path followed (i.e. the values of the contextual elements are required for selecting a branch, i.e. an element of reasoning among several ones). Figure 3 provides the definition of the elements in a contextual graph. A more complete presentation of this formalism and its implementation can be found in [4].



1: Contextual element (circles)
branch-1 2: Activity (ovals)
branch-2 3: Temporal Branching (vertical lines)
branch-1 4: Action
branch-2 5: Action
branch-3 6: Action

Fig. 3. Elements of a contextual graph

Elements of a contextual graph are: actions, contextual elements, sub-graphs, activities and temporal branchings.

- An **action** is the building block of contextual graphs. We call it an action but it would be better to consider it as an elementary task. An action can appear on several paths. This leads us to speak of instances of a given action, because an action, which appears on several paths in a contextual graph, is considered each time in a specific context.
- A **contextual element** is a couple of nodes, a contextual node and a recombination node; A contextual node has one input and N branches [1, N] corresponding to the N instantiations of the contextual element already encountered. The recombination node is [N, 1] and shows that even if we know the current instantiation of the contextual element, once the part of the practice on the branch between the contextual and recombination nodes corresponding to a given instantiation of the contextual element has been executed, it does not matter to know this instantiation because we do not need to differentiate a state of affairs any more with respect to this value. Then, the contextual element leaves the proceduralized context and (globally) is considered to go back to the contextual knowledge.
- A **sub-graph** is itself a contextual graph. This is a method to decompose a part of the task in different way according to the context and the different methods existing. In contextual graphs, sub-graphs are mainly used for obtaining different displays of the contextual graph on the graphical interface by some mechanisms of aggregation and expansion like in conceptual graphs [18].
- An **activity** is a particular sub-graph (and thus also a contextual graph by itself) that is identified by actors because appearing on different paths and/or in several contextual graphs. This recurring sub-structure is generally considered as a complex action. An activity is similar to a scheme as considered in cognitive ergonomics [13]. Each scheme organizes the activity around an object and can call other schemes to complete specific sub-goals.

- A **temporal branching** expresses the fact (and reduces the complexity of the representation) that several groups of actions must be accomplished but that the order in which action groups must be considered is not important, or even could be done in parallel, but all actions must be accomplished before continuing. The temporal branching is for context what activities are for actions (i.e. complex actions). This item expresses a problem of representation at a lower granularity. For example, the activity "Make train empty of travelers" in the SART application [14] accounts for the damaged train and the helping train. There is no importance to empty of travelers first either the damaged train or the helping train or both in parallel. This operation is at a too low level with respect to the general task "Return back rapidly to a normal service" and would have otherwise to be detailed in the three paths in parallel leading to the same sequence of actions after.

Some mechanisms of aggregation and expansion provide different local views on a contextual graph at different levels of detail by aggregating a sub-graph in an item (a temporary activity) or expanding it. This representation is used for the recording of the practices developed by users, which thus are responsible for some paths in the contextual graph, or at least some parts of them.

3.2 An Explanation Typology Established from Contextual Graphs

We established a typology of explanations, based on previous works and exploiting the capabilities of contextual graphs [2]. By adding a new practice, several contextual information pieces are recorded automatically (date of creation, creator, the practice-parent) and others are provided by the user himself like a definition and comments on the item that is introduced. Such contextual information is exploited during the explanation generation. Thus, the richness of contextual-graph formalism leads in the expressiveness, first, of the knowledge and reasoning represented, and, second, of the explanations addressing different users' requirements. The main categories of explanations identified in contextual graphs are:

- **Visual explanations** correspond to a graphical presentation of a set of complex information generally associated with the evolution of an item, e.g. the contextual graph itself, the decomposition of a given practice, the series of changes introduced by a given user, regularities in contextual graphs, etc.

- **Dynamic explanations.** They correspond to the progress of the problem solving during a simulation addressing questions as the "What if" question. With the mechanisms of aggregation and expansion, a user can ask an explanation in two different contexts and thus receives two explanations with different presentations (e.g. with the details of what an activity is doing in one of the two explanations). The dynamic nature of the explanation is also related to the fact that items are not introduced chronologically in a contextual graph. For example, in Figure 2, the contextual element 5 (Duration of the download?) was introduced after the contextual element 6 (Is it for a course?). Initially, the user was retrieving information on web pages, and one day he arrives on a Web site particularly slow and observes that he did not wait more than a couple of seconds. Then, the user decides to add the duration of the download as an important contextual element. Finally, the proceduralized context along a practice is an ordered series of instantiated contextual elements, and changing the instantiation of one of them is changing of practice and thus changing of explanation.

- **User-based explanations**. The user being responsible of some practice changes in the contextual graph, the system uses this information to tailor its explanation by detailing parts unknown of the user and sum up parts developed by the user. Such an explanation allows the author of a practice to identify the contextual elements that he had not taken into account initially and that has been introduced by other users).
- **Context-based explanations.** The definition of the proceduralized context (an ordered sequence of instantiated contextual elements) shows that a given item (say the activity represented by an oval in Figure 2) appearing on different branches of the contextual graph, appear in different contexts. This means that any explanation of the activity cannot be the same on the three branches. We exploit this finding in our driver-modeling application for representing "good" and "bad" behaviors on the same contextual graph [6]. Thus a relevant explanation relies heavily on the building of the proceduralized context, and because the contextual graph can be incrementally enriched, explanations can be richer also.
- **Micro- and macro-explanations**. Again, with the mechanisms of aggregation and expansion, it is possible to generate an explanation at different levels of detail. For such a complex item like an activity or a sub-graph, it is possible to provide on them a micro-explanation from an internal viewpoint on the basis of activity components. A macro-explanation from an external viewpoint is built with respect to the location of the activity in the contextual graph like any item. This allows to providing (at least) two different types of explanation on the activity "Make your train empty of travelers" at the macro level of the subway-line responsible and at the micro level of the train driver. Note that the subway-line responsible may ask the micro-explanation in case of doubt on one operation of the driver. This twofold explanation is linked to the notion of activity, but can be used by any user with aggregation and expansion of local sub-graphs of parts of the whole contextual graph.
- **Real-time explanations**. There are three types of such explanations. First, the explanation is asked during a problem solving when the system fails to match the user's practice with its recorded practices. Then, the system needs to acquire incrementally new knowledge and to learn the corresponding practice developed by the user (generally due to specific values of contextual elements not taken into account before). This is an explanation from the user to the machine. Second, the user wished to follow the reasoning of a colleague having solved the problem with a new practice (and then we are back to simulation). Three, the system tries to anticipate the user's reasoning from its contextual graph and provides the user with suggestions and explanations when the user is operating. These suggestion and explanation rely on the contextual elements that are explicitly considered in the contextual graph. Note that it is because the system

fails to represent a user's practice that the user explains to the system the new practice by introducing new knowledge, knowledge that the system can reuse after.

Moreover, these different types of explanation (and others that we are discovering progressively) can be combined in different ways such as visual and dynamic explanations.

4. Conclusion

Relevant explanations are a crucial factor in any collaboration between human actors, especially when they interact by computer-mediated means. First, collaboration looses some advantages of a face-to-face collaboration in which a number of contextual elements are exchanged in parallel with the direct communication. Second, collaboration can benefit of new ways to replace this "hidden exchanges" of contextual cues between actors by the use of the computer-means themselves.

Explanation generation is very promising for collaboration because explanations use and help to maintain a shared context among actors. We are now in a situation in which computer-mediated interaction concerns human and software actors. Software must be able to react in the best way for human actors. For example, for presenting a complex set of data, a software piece could choose a visual explanation taking into account the type of information that human actors are looking for. We show that making context explicit allows the generation of relevant explanations. Conversely, explanations are a way to make contextual knowledge explicit and points out the relationships between context and the task at hand, and thus develop a real shared context.

In this paper, we argue that a key factor for the success of relevant explanations is to use a context-based formalism, like Contextual Graphs, that represent in a uniform way all the richness of the knowledge and reasoning in the focus. A good option is to consider context of use simultaneously with the knowledge. As a consequence, this allows developing new types of explanation like visual explanations, dynamic explanations, real-time explanations, etc. Indeed, we have developed a new typology of explanations that include past works on explanations but goes largely beyond. Moreover, these different types of explanations can be combined together to provide richer explanations.

However, this is only the first step. A promising path is to explore intelligent assistant systems. Indeed, computer-mediated means can keep and reuse a trace of interaction between human actors. In real-time situations, the human actor cannot loose time to answer questions of a machine because the actor is generally under time pressure (e.g. an incident solving in a control room), but the machine can act in parallel with actors in a kind of personal simulation replaying similar past situations, and making suggestions when appropriate. In that sense, the machine may become an excellent secretary, fixing alone all the simple problems of human actors, and preparing a complete folder on complex situations letting actors make their decision. Here, the machine generates explanations for humans.

Conversely, when the machine fails to address correctly a problem, the machine may benefit of its interaction with the human actors to acquire incrementally the missing knowledge and learn new practices. As a consequence, the machine will be able to explain later its choices and decisions. Now, there is a software piece called Contextual Graphs that is able to manage incremental acquisition and learning, and begins to provide some elementary explanations.

As a general learned lesson, expressiveness of the knowledge and reasoning models depends essentially of the representation formalism chosen for expressing such models. This appears a key element of collaboration with multiple sources of knowledge and different lines of reasoning intertwined in a group work. This is a partial answer to our initial observation that collaboration would be better understood if we consider jointly its two dimensions, the human dimension and the technology dimension. Then, explanation generation would be revised in order to develop "collective explanations" for all the (human) participants in the collaboration, that is in each mental representation.

References

1. Abu-Hakima S. et Brézillon P.: *Knowledge acquisition and explanation for diagnosis in context*, Research Report 94/11, LAFORIA, University Paris VI, Paris, France (1994)
2. Benci, P., Brézillon, P. and Potier, D.: Acquiring and representing users' practices by contextual graphs. Brevia and Demonstrastions Presentations associated with the 5th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-05), Paris France. LIP6 2005/007. With A. Dey, B. Kokinov, D. Leake, R. Turner (2005)
3. Brézillon, P.: Context modeling: Task model and model of practices. Proceedings of the 5th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT-07, A. Dey, B.Kokinov, D.Leake, R.Turner (Eds.), Springer Verlag, LNCS 3554, pp. 55-68 (2007)
4. Brézillon, P.: *Task-realization models in Contextual Graphs*. In 5th International and Interdisciplinary Conference on Modeling and Using Context, Lectures Notes in Artificial Intelligence, Vol 3554, pp. 55--68, Springer-Verlag (2005)
5. Brézillon, P.: Focusing on context in human-centered computing *IEEE Intelligent Systems*, 18(3): 62-66 (2003)
6. Brézillon, P. and Brézillon, J.: Contextualized task modeling. Revue d'Intelligence Artificielle (2008)
7. Brézillon, J. and Brézillon, P.: Context modeling: Context as a dressing of a focus. Proceedings of the 5th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT-07, LNAI 4635, Springer Verlag, pp. 136-149 (2007)
8. Brézillon P. et Pomerol J-Ch.: *Contextual Knowledge sharing and cooperation in intelligent assistant systems*, Le Travail Humain 62 (3), PUF, Paris, 223-246 (1999)
9. Brézillon, P. and Pomerol, J.-Ch. :Lessons learned on successes and failures of KBSs. Special Issue on Successes and Pitfalls of Knowledge-Based Systems in Real-World Applications. Failures and Lessons Learned in Information Technology Management, June, 1(2), pp. 89-98 (1997)
10. Karsenty L. et Brézillon P.: *Cooperative problem solving and explanation*, International Journal of Expert Systems With Applications, 8(4): 445-462 (1995)

11. Kock, N., Davison, R., Ocker, R. & Wazlawick, R.: E-Collaboration: A look at past research and future challenges. Journal of Systems and Information Technology, 5(1), 1-9 (2001)
12. Kodratoff, Y.: Is artificial intelligence a subfield of computer science or is artificial intelligence the science of explanation? Sigma Press, Bled. Progress in Machine Learning,pp. 91-106 (1987)
13. Leplat, J. and Hoc, J.M.: Tâche et activité dans l'analyse psychologique des situations. Cahiers de Psychologie Cognitive, 3, 49-63 (1983)
14. Pomerol, J.-Ch., Brézillon, P. and Pasquier, L.: Operational knowledge representation for practical decision making. Journal of Management Information Systems, 18(4): 101-116 (2002)
15. PRC-GDR: Actes des 3^{e} journées nationales PRC-GDR IA organisées par le CNRS, textes réunis par Bernadette Bouchon-Meunier, Chapitre 7, Editions Hermes (1990)
16. Richard, J.F.: Logique du fonctionnement et logique de l'utilisation. Rapport de Recherche INRIA no 202 (1983)
17. Schank R.C.: *Dynamic memory, a theory of learning in computers and people*, Cambridge University Press (1982)
18. Sowa J.F.: *Knowledge Representation : Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA (2000)
19. Spieker, P.: *Natürlichsprachliche Erklärungen in technischen Expertensystemen*. Ph.D. Dissertation, University of Kaiserslautern, (1991)
20. Swartout, W.R. and Moore, J.D.: *Explanation in second generation expert systems*. In J. David, J. Krivine, and Simmons, R. (Eds.): Second Generation Expert Systems, pages 543 585, Berlin, Springer Verlag (1993).