Proc. of the European Conference on Cognitive Science, Manchester, UK, April 1997, pp. 129-139.

JOINT COGNITIVE SYSTEMS, COOPERATIVE SYSTEMS AND DECISION SUPPORT SYSTEMS: A COOPERATION IN CONTEXT

by P. BREZILLON[1] and J.-Ch. POMEROL[1]

ABSTRACT
We present the lessons drawn from a review of the main concepts put forward by the designers of decision support systems, joint cognitive systems and cooperative systems. A striking observation is that these systems stumble on user-system cooperation. The main idea of this paper is that interactive system must behave more as an intelligent assistant than as an expert. Key elements of an effective cooperation between a user and a system are making explicit cooperation context and extending consequently the notions of explanations, knowledge acquisition and machine learning.

Keywords: *Decision support systems, joint cognitive systems, cooperative systems, intelligent assistant systems, cooperation, context, explanation, incremental knowledge acquisition.*

I. INTRODUCTION

This paper considers users that have to make a decision. Such users are either decision makers or operators controlling a process (whatever this process is). In both cases, users have the responsibility of the final decision. Thus, user, decision maker and operator will be used interchangeably.

The word Decision Support System (DSS) was launched about twenty-five years ago by Scott Morton in reaction against the many operational information systems suitable for executing routine tasks while they were unable to support a decision maker in more skillful jobs. During the seventies, DSS concepts evolved from functional definitions: a system that "improves the effectiveness of the decision making rather than its efficiency" (Keen and Scott Morton, 1978) to more descriptive and formal definitions (Bonczek et al.,1981; Lévine and Pomerol, 1989). For instance, Bonczek et al. (1981) stress the view of a DSS as a problem solver, approximately in the AI sense, whereas Lévine and Pomerol (1989) define a DSS as a particular Information Processing System designed to permit a heuristic search in both spaces of data and models.

The DSS concept has known some further evolution towards EIS (Executive Information Systems) in which models are simplified whereas the user-friendliness is increased and now to multidimensional spreadsheets and data-mining tools. Meanwhile, and almost separately AI moved from the view of expert systems replacing people to Joint Cognitive systems (JCSs) working in a symbiotic manner with the users (Woods et al., 1990). This evolution was also visible in the book of Winograd and Flores (1986) in which the last chapter is an illustration and a defense of interactivity, very similar to that put forward by DSS designers ten years ago and since ever. For example, Keen and Scott Morton (1978, p-2) wrote that "The relevance for managers is the creation of a supportive tool, under their own control, which does not attempt to automate the decision process, predefine objectives, or impose solutions."

[1] LIP6, Case 169, University of Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France

Indeed, the introduction of the JCS concept is a consequence of the weaknesses of KBSs as perceived by the IA community. Brézillon and Pomerol (1996 and 1997) stress in a recent review that the main problems with KBSs are at the level of human-machine interaction. Most of the functions of a system must be revised at this level. This concerns knowledge acquisition, machine learning, explanation generation, and the recent consideration in AI of context (Brézillon, 1996).

We will show that the JCS concept mainly emerges as a new paradigm of Knowledge Based Systems (KBSs), and actually merges different currents. Even if generally not acknowledged, the JCS concept is very close to DSS ideas. Our main purpose in this paper is to compare DSS and JCS approaches. We assume that the reader has a good understanding of the DSS ideas developed in the literature and synthetic treatises as (Keen and Scott Morton, 1978; Bonczek et al., 1981; Sprague and Carlson, 1982; Lévine and Pomerol, 1989). Thus we begin with the most recent works about JCS, trying to extract their main features and to underline the usual views about interactivity in AI. Second, we will compare these features with the usual DSS views. Third, we will finish by a synthesis of the main characteristics shared by supportive tools issued either from DSS or AI cultures and we will show their convergence towards the paradigm of intelligent assistant system.

II. JOINT COGNITIVE SYSTEMS

Woods et al. (1990) coined the expression "Joint Cognitive System" to stress the fact that neither the system nor the user is able to solve the problem at hand alone. An important word in *Joint Cognitive Systems* is the word "cognitive" that gives an emphasis on the cognitive aspects of task solving. In a broad sense, this implies that the system and the user are able to understand each other when solving a problem. This may be regarded as being utopic because it is unlikely that a machine understands users. In a metaphoric sense however, it entails some very strong requirements about the design of the system as: the sharing of the cognitive representations, an agreement about the contextual knowledge of the problem solving, the ability to follow each other reasoning, the capability of exchanging explanations, and a sharing of the interaction control.

The question is not whether the **sharing of cognitive representations** is desirable or not, but how to achieve it. This requirement would be fulfilled when users' cognitive representations are disclosed and implemented in the system. However, in many human tasks it is not obvious to get a comprehensive view of the representations at work in the subject brain because things intervening in human reasoning may be not suitable for usual representation (e.g., see the somatic markers in decision tasks, Damasio, 1994, and Pomerol, 1997).

A more realistic approach is to provide means for the user and the system to make their interpretations compatible (Karsenty and Brézillon, 1995). Each one builds his interpretation of an observation with his own cognitive representation. This is particularly important because the user's cognitive representation is richer than that of the system. In case of a divergence in their interpretations, they can compare and subsequently update the contextual elements on which rely their interpretations. Such a system must be able to incrementally acquire knowledge and provide explanation in the cooperation context. This point stresses the need to revise our view on human-machine cooperation because the cooperation takes place in a context in which elements must be shared.

The requirement about the **contextual knowledge** of the problem solving is also a matter of representation. Recent studies emphasize the fact that contextual information is a necessary component for the achievement of any task (see Brézillon and Abu-Hakima, 1994; Brézillon, 1996). This implies that the designer must represent context as computer data to make it accessible to the system and shared by the system and the user.

A difficulty is the infinite dimension of context (McCarthy, 1993) whereas designer's resources are limited so that the developer has to draw a frontier between the data that are used in the problem solving, the contextual data that only constrain the problem solving, and the knowledge that may be left outside of the system. This is not a flat problem because the nature of these three types of data change from one step of the problem solving to the next one. A piece of data may be contextualized at one step of the problem solving, contextual at the following one and left outside after (Brézillon, 1997).

This raises the difficult questions of the limits of a system. What does the system know? In what context is it safe to use it or not? This last question is very important to prevent accidents (Boy, 1991; Hoc, 1996). Indeed, context can only be studied through its use (Brézillon et al., 1997). This limits some formal approaches because context cannot be studied separately from its operational use (Brézillon, 1996).

System capabilities of **explaining its reasoning** to a user is a well studied issue. However, few systems are able to provide explanations that are relevant for the user. This is true for stand alone systems, and crucial for joint cognitive systems. The main reason is that a feedback from the user is insufficient to generate relevant explanations: explanations must be co-built by the system and the user (Karsenty and Brézillon, 1995). Explanation needs occurring during the problem solving, explanation generation must be treated as an intrinsic part of any cooperative problem solving. One must explain to cooperate and cooperate to explain.

Another important problem in human-machine cooperation is that it is also crucial that the system can accept explanation from the user. Again, a co-building of the explanations may bring a solution to this problem. As a consequence, for accepting an explanation, the system must be able to incrementally acquire knowledge during the problem solving. This implies to consider cooperation in a broad view.

Information exchange and man-machine communication are also necessary to control the exploration among the many informational states of the system. Actually, the user performs a heuristic search among many informational states (Lévine and Pomerol, 1989, 1995). This exploration obeys the usual rules of any heuristic search and must be controlled by the user. Practically, this implies from the user's viewpoint to account for notions as interuptability, reorientation, resuming facilities at any point, information requests on the status of the search (Where are we? What can be done from this state?). From the system's viewpoint, this implies an intelligent assistant behavior. Thus, the system must be able to propose satisfying alternatives and argue on them, recall points of the past interaction, have a model of the user and a model of the process on which the user work. This defines a particular type of cooperation (with an asymetry in the respective roles) between the user and the system in which the information sharing plays an important role.

Putting the accent on cooperation means that the system and the user complement each other. They need the support of the other to achieve their common task. This claim is not so obvious because, when it is possible to automate a system, the automatic system works quite well. When it is not possible, it is often better to let people continue to handle the task without any computer aid, or at least as an intelligent assistant system. Thus there is a compromise to find on the basis of who must take the final decision, what the importance of the task is, the degree of seriousness of the situation.

III. COOPERATIVE SYSTEMS

Cooperation is considered in different situations in which intervene two humans, two software agents, or a human and a machine. Through the literature, various elements are evoked to specify cooperation: decomposition and allocation of tasks, coordination, collaboration, organization, negotiation, conflict management, initiative, control, dealing with alternative solutions, etc. Generally, any combination of some of these elements leads to a cooperation model that depends on the nature of the application. For human-machine cooperation, all these elements do not present the same importance, especially when the user will make the final decision.

Cooperative systems are based on a successful combination of human skills and computing power. The goal is to construct systems that augment and amplify human intelligence in problem solving, decision making and information management rather than replace it. Such systems must be active agents working in real-time and in synchronization with the user (Clarke and Smyth, 1993). The main virtue of the word "cooperative" for us is the emphasis on: (1) the complementarity of the system and the user, (2) the communication chanels between the human and the machine, and (3) the dynamical aspect of the complementarity of this process and its dependence on the context of the interaction.

User-system complementarity implies that the system may contain either some knowledge (model or data) or functions (e.g., database exploitation) that really complements the user's skills. This is a perspective that was already discussed in the DSS field where people generally advocate for normative models complementing and being "more rational" than the user. The asymmetry of the agents, i.e. the user and the system, must be exploited. For instance, the system can compute large quantities of calculus (e.g., probabilities), and the user may produce interpretations of data provided by the system (Fischer, 1990). This implies creating an environment where the refinement of solutions can be based on argument and the resolution of differing viewpoints (Clarke and Smyth, 1993).

However, some researchers point out the lack of confidence of the user in such models and his reluctance to use them (e.g., see Keen, 1982). The question of the user confidence in the system is crucial because there is no cooperation without confidence. It is noteworthy that the complementation of the user by some models, different from those that would naturally be implemented by the user, is antinomic with representation sharing advocated by JCS designers.

In order to take advantage of their complementarity, the human and the machine must have efficient means of communication. Effective **human-computer communication** requires providing the system with a considerable body of knowledge about the world to agree on terms of reference to be adopted during the process. Establishing this "common ground" in the form of shared understanding and agreement between them is a crucial aspect of communication. (Knowing what to not communicate also is important.)

There are two elements of communication that are not often evoked in the literature, namely explanation and context, the later being discussed in the next section. In human-machine cooperation, the machine and the user have to explain. The role of an explanation is to persuade the other agent that a partial solution is correct, not just to provide a final justification of an already determined completed solution. For instance, explanations provide some information about: (1) what the system can or cannot do; (2) what the system has done; (3) explain what the system is trying to do; (4) why it is doing what it does by responding to the user's clarification questions. Explanations aim to modify the knowledge of the other agent (bringing new pieces of knowledge or revising existing ones); to clarify the context of the shared knowledge; to convey new information pieces unknown by the explainee; to reach a common agreement; to coordinate the agents' activities; to rethink our own implicit assumptions, forcing us to trace our own reasoning process and revealing alternative ways to think about a problem.

The relationship between **context and complementarity** is linked to the dynamic aspect of the cooperation process. During such a process, the problem is progressively jointly solved by the user and the system. May be each of these actors have some personal final goal, but they need each other to achieve intermediary goals. Thus, user-system cooperation implies an evolution of the system during cooperation because the system has more limited capabilities than the user and can learn from its failures. This also implies to consider the system evolution on many solving: learning from its failures on one problem solving, the system will be more efficient in other ones after. The most important point here is that the system will acquire the needed knowledge in its context of use. This is, in our opinion, one of the main weaknesses of first expert systems that acquired knowledge before to use it (Brezillon and Pomerol, 1996).

The system must be able to learn during the progress of the problem solving and understands the changes introduced by the user's interventions. The two main points here are the ability to accept explanation and the incremental acquisition of knowledge. In such a situation, context is not an abstracted item, but corresponds to a concrete thing. However, this rises again the question of the different types of context: the cooperation context, the reasoning context and the context in which knowledge is represented. In all the cases, accounting for the cooperation evolutivity implies to make the cooperation context explicit.

Some authors have put forward the point that cooperation implies a sharing of goals. Especially in JCSs, Woods et al. (1990) said that the system is goal-driven. This is not so obvious, because cooperation in the world may occur and occurred between people having different goals and knowing it (remember for example, Staline and Hitler for a while, then Staline and the capitalistic countries, and so on). Common intermediate subgoals suffice for cooperation even if each agent has personal goals different of those of the other.

For instance, consider an operator that has in charge a real-world process, say in a nuclear plant application as discussed by Brézillon and Cases (1995). The operator's goal is to ensure correct operating of the process. The system's "goal" is to assist intelligently the operator. To do this, the system tries to act in parallel with the operator, and using its means of simulation, to predict the future behavior of the process. This cooperation may then take different forms: a waking state if its prediction is compatible with the operator's actions and the process's behavior, and a cooperative state if a difference is noted on either the operator's action or his consequences in the future. The goal of an intelligent assistant system is to prevent operator's actions that may be a problem soon or later. Hoc (1996) also evoked the problem when exists a delay between an action and its consequences (about the case of an action on the helm of a boat and its evolution).

Here, the most important insights introduced by the JCS view are: the complementary between the system and the user, the co-evolution of the system and the user during problem solving, and co-solving of the problem. Comparing these views with previous ones as the DSS view, one sees that there is no real difference between all the types of system. They all raise the question of providing a computer support to users carrying out various tasks. When they arrive to practical conclusions, these are almost similar and certainly useful. These recommendations are centered at defining a good interactivity: explanations, representation sharing, explicit context, user control, etc.

Besides these very involved problems of user-system communication and interactivity, it remains the question of the complementarity between the user and the system. What are the models and data really useful for a task? Can the system impose some models if they are in some sense, better than those of the user? In this case, how can one maintain interactivity and cooperation? Will a user accept a system using models that he does not understand or he is not confident in? All these questions are not new and were more or less already discussed in DSS literature.

IV. LESSONS LEARNED FROM THE THREE APPROACHES

IV.1. INTRODUCTION

The main differences among JCS, cooperative system and DSS views are differences of accentuation. The JCS literature is centered on cognitive representation. The cooperative-system literature focus on the necessary complementarity between the system and the user. The DSS researchers are more interested in modelling and development methodologies. However, they all arrive to the same practical conclusions about the necessity of cooperation. They do not fundamentally diverge about representations and models, but the practical issues in this domain are very dependent on the task considered and actually few general ideas emerge.

DSSs, JCSs, Cooperative Systems address the same issue: how a computer system and human being can, jointly, in real time, cooperate to the achievement of a task, so that the resulting achievement be better than if it was carried out by the system or the user alone? The human and computer weaknesses and strengths are quite well studied (e.g., in the decision field, von Winterfeldt and Edwards, 1986, Tversky and Kahnemann, 1988). Nevertheless, the numerous failures (Brézillon and Pomerol, 1996, 1997), the lack of a solid core of explanations of these failures paves the way for a continuous quest of new paradigms. Also, most systems whether they succeed or fail are rather contingent and unique. This makes it difficult to settle a scientific doctrine relying on solid renewable experiences.

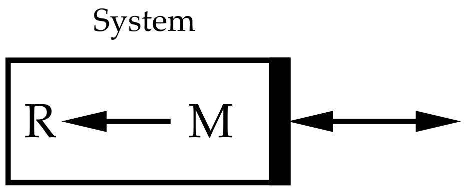
It appears that some elements are missing in the three approaches: the context of the cooperation, the integration of explanations in the cooperation, and the importance of incremental knowledge acquisition. We develop in this section these three aspects.

IV.2. THE CONTEXT OF COOPERATION

Context plays an important role in a number of domains since a long time. This is especially true for activities as predicting context changes, explaining unanticipated events and helping to handle them, and helping to focus attention. In Artificial Intelligence, context was first introduced in a logicist framework by McCarthy (McCarthy, 1993). However, there is no clear definition of this concept (Brézillon, 1996): a set of preferences and/or beliefs; a window on the screen; an infinite and only partially known collection of assumptions; a list of attributes; the product of an interpretation, and a collection of context schemas; paths in information retrieval; slots in object-oriented languages; buttons which are functional customisable and shareable; possible worlds; assumptions under which a statement is true or false; a special, buffer-like data structure; an interpreter which controls the system's activity; the characteristics of the situation and the goals of the knowledge use; or entities (things or events) related in a certain way that permits to listen what is said and what is not said.

Indeed, the notion of context is dependent on its interpretation on a cognitive science viewpoint versus an engineering (or system building) viewpoint. The *cognitive science view* is that context is used to model interactions and situations in a world of infinite breadth, and human behavior is the key in extracting such a model. The *engineering view* is that context is useful in representing and reasoning about a restricted state space within which a problem can be solved [Brézillon and Abu-Hakima, 1995].

The different definitions of context can be discussed on the following figure inspired by the general structure of an information processing system proposed by Newell and Simon (1972):



In this figure, the system is supposed to interact with an agent called A (the black part of the box system represents the interface, but will be not discussed here). The agent may be a human being, a computer system or any combination of them. The system itself is composed of an inference engine and a knowledge base on the basis of an enlarged view: a mechanism (M) for reasoning and a repository (R). The repository R may contain knowledge, data, information, domain and task models, ontologies, etc. R is supposed to be controlled by M that manages items in the repository, introduces items in it and retrieves items from it. In the terms of Newell and Simon, R is the Memory, M the Processor, and the interface part, the Effectors and Receptors.

According to this model, five types of context may be pointed out at the level of each entity R, M, A, and the two arrows. For example, the cooperation context is at the level of the double arrow. More complex contexts (e.g., for case-based reasoning) are combinations of several elementary contexts (e.g., contexts of R and M in case-based reasoning). This model permits to conciliate opposite views on context, as the static vs. dynamic aspects of context, the discrete vs. continuous nature of context, the relationships between contextual knowledge and contextualized knowledge. It is also pointed out that these different types of context are not independent [Brézillon, 1996]. Hereafter, we focus on the cooperation context.

Cooperation context is the knowledge shared by the agents in a cooperation at each step of the problem solving (Cahour 93). Cooperation implies a transfer of knowledge for the problem solving and the transfer of contextual information between agents. The cooperation context contains items like the dialogue memory, the task at hand, the spatio-temporal situation (place, moment, elements, etc.), and the psycho-social situation (user model, roles and social status, etc.). Contextualizing the cooperation has the advantages of: (i) reducing the explainer's effort and allowing to reach a point of mutual understanding more rapidly if the answer confirms the explainee's assumption; and (ii) allowing the explainer to predict an explanation need if the answer differs from the explainee's assumption.

The main point is that the cooperation context evolves dynamically along the problem solving when an item is introduced by one agent, or when both agents agree on a step of the problem solving and move to the next step.

IV.3. EXPLANATIONS IN COOPERATION

Karsenty and Brézillon (1995) show that explanation is an intrinsic part of cooperation. On the one hand, explanations improve the cooperation, and, on the other hand, cooperation permits to each agent to produce relevant explanations for the other. An important result is that a system must be able to accept explanations from the user, and then change its knowledge organization accordingly. Explanations is useful to allow the integration in the knowledge of one agent of some pieces of the other's knowledge (Brézillon, 1994). As a consequence a system must be able to acquire incrementally knowledge pieces and their context of use. Explanations are a way to facilitate the acquisition of the missing knowledge that must be acquired when needed.

Produced spontaneously, explanations are also a means for anticipating the explanation needs of the other. They permit to enrich the context that is shared by both agents, leading to a mutual understanding and a mutual acceptance, and thus facilitating their cooperation. Spontaneous explanations have numerous advantages: they make understanding easier, reduce misunderstandings, insure that the interpretation of the agents' solutions was the one they wanted to communicate, they decrease the number of dialogue exchange needed to achieve the interaction goal, they increase the shared knowledge. The main goal of spontaneous explanations is the validation of the cooperation context. However, a system would have to decide when it is necessary to volunteer explanations.

Explanation and context are thus intertwined aspects of cooperation. An explanation changes the current state of the cooperation context, and making context explicit is a way to provide explanations to the other.

IV.4. INCREMENTAL KNOWLEDGE ACQUISITION

The generation of explanations by the user to the system implies that the system may incrementally acquire knowledge, assimilate it in its knowledge base, and propagate the consequences of the new knowledge. Capturing and using knowledge in the context of use greatly simplifies knowledge acquisition because knowledge provided by experts is always in a specific context and is essentially a justification of the expert's judgment in that context. By incrementally acquiring knowledge, the system improves its performance over time.

Through the user-system cooperation and the explicit definition and use of context, new knowledge is acquired and validated. This knowledge is acquired through clarification dialogue. If the system can identify gradually the target knowledge, it can verify if this knowledge is missing or not. Incremental knowledge acquisition plays an important role in two situations:

- First, the knowledge missing in the current context must be acquired when needed in order to permit the cooperation to resume. For missing knowledge in the system, the user should be provided with the capability to add new knowledge through explanations. Here, explanations permit to make explicit the context of the knowledge use. In the case of interactive knowledge acquisition through explanation, a user introduces a pseudo-formal specification for a change in the knowledge that is evaluated with respect to the other knowledge in the system.
- Second, a chunk of knowledge may be known by the system but used incorrectly, i.e., a link for the current context is missing. This link could be added in the form of a consequence of the current context. Here, incremental knowledge acquisition will focus on the refinement of the context of use of the chunk of knowledge, and explanations must support such a refinement.

Incremental knowledge acquisition and symbolic learning are two complementary dimensions of a space in which a system may enrich its knowledge. The vertical axe would represent symbolic learning that is a process of abstraction and generalization starting from examples to identify concepts. The horizontal axe would be incremental knowledge acquisition that aims at integrate new pieces of knowledge in already organized knowledge.

Such a system will start with the kernel of knowledge that has been entered manually, and increase its level of assistance in acquiring and refining that knowledge, both in terms of the increased level of automation in interacting with users, and in terms of the increased generality of the knowledge. Such a system is at the intersection of machine learning and knowledge acquisition [Baudin et al., 1994; Bocionek, 1995]: it is a first step towards a system that moves along a continuum from interactive knowledge acquisition to increasingly automated machine learning as it acquires more knowledge and experience.

V. CONCLUSION

Damasio (1994, 1996) quoted the "lack of sensitive experience of the computers." There are two important words in this expression: "sensitive" and "experience." We do not expect that computers will acquire some sensitivity in future, but they can gain experience by learning algorithms. The first objective is then to include learning capabilities. Learning is a key capability for a better coopération, but it is also crucial to alleviate the data acquisition burden and to apprehend contextual information. However, machine learning capabilities are not sufficient, a system must also be able to incrementally acquire knowledge from the user.

Indeed, cooperation, explanation, incremental knowledge acquisition, machine learning and context appear to be only different facets of intelligent assistant systems (Boy, 1991; Brézillon and Cases, 1995). An intelligent assistant system must: maintain interaction with its environment (observing in the world and performing actions in response, reactiveness); be able to take the initiative (pro-activeness); be able to perform social actions (communication, cooperation); work autonomously in the background; react to incoming messages; make decisions on its user's behalf as often as possible; and be able to generate goals independently and act rationally to achieve them (planning and plan execution, autonomy). Such a system should not only support single tasks, but assist their users in much the same way as human secretaries do. For example, such programs might assist a worker in scheduling meetings, flagging important electronic mail, processing purchase orders, etc. In our understanding, three aspects are particularly important for developing such intelligent assistant systems:

(1) Explanation. The role of the explanations in a cooperative problem solving must be revised, because explanation is an intrinsic part of any cooperation (Karsenty and Brézillon, 1995).
(2) Incremental knowledge acquisition. The system needs to incrementally acquire knowledge from user to after relieve users with the same type of problems (Abu-Hakima and Brézillon, 1994).
(3) Context. It is necessary to make the context of the user-system cooperation explicit in the problem solving to provide relevant explanation and acquire incrementally knowledge (Brézillon and Abu-Hakima, 1995).

Our team is developing two intelligent assistant systems for monitoring subway operations (Brézillon et al., 1997) and in Enology (Agabra et al., 1997), thanks to a multi-agent approach. Each function of a system is ensured by a subsystem called an agent. In the first application, one of the agent is an incident manager. It records the characteristics of an incident, the context in which the incident occurred, and the strategy used for solving the incident. In the second application, an agent tackle expertise coming from different human expert, having each a partial view of the domain, for supporting the enologist in prediction of fermentation stops.

REFERENCES

Agabra, J., Alvarez, I., & Brézillon, P. (1997) Contextual knowledge based system: A study and design in enology. *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context* (CONTEXT-97). Federal University of Rio de Janeiro (Ed.), pp. 351-362.

Baudin, C., Pell, B., and Kedar, S. (1994) Increasing levels of assistance in refinement of knowledge-based retrieval systems. *Knowledge Acquisition*. 6, pp. 179-196.

Bocionek, S.R. (1995) Agent systems that negotiate and learn. *International Journal of Human-Computer Studies*. 42(3): 265-288

Bonczek, R.H., Holsapple, C.W., & Whinston, A.B. (1981). *Foundations of Decision Support Systems*. Academic Press.

Boy, G. (1991). *Intelligent Assistant Systems*. Academic Press.

Brézillon, P. (1994) Context needs in cooperative building of explanations. *Proceedings of the First European Conference on Cognitive Science in Industry*. Centre de Recherche Public, Luxembourg, pp. 443-450.

Brézillon, P. (1996). *Context in human-machine problem solving: A survey* (Research Report 96/29, LAFORIA, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France, October, 37 pages).

Brézillon, P. (1997). A first step towards the modeling of contexts. Proceedings of the ECCS97 Workshop on Context held jointly with the 2nd European Conference on Cognitive Science, Manchester, UK, April 1997 (to appear).

Brézillon, P., & Abu-Hakima, S. (1994). *Knowledge acquisition and explanation for diagnosis in context* (Research Report 94/11, LAFORIA, University Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France).

Brézillon, P., and Abu-Hakima, S. (1995). Using Knowledge in its context: Report on the IJCAI-93 Workshop. *AI Magazine*. 16(1): 87-91.

Brézillon, P. & Cases, E. (1995). "Cooperating for assisting intelligently operators", Proceedings of the International Workshop on the Design of Cooperative Systems, Antibes, France, January 1995, pp. 370-384.

Brézillon, P., & Pomerol, J.-Ch. (1996). Misuse and Nouse of Knowledge-Based Systems: the past experiences revisited, In P. Humphreys, L. Bannon, A. McCosh, P. Migliarese, & J.-Ch. Pomerol (Eds.), *Implementing Systems for Supporting Management Decisions* (pp.44-60). Chapman and Hall.

Brezillon, P., & Pomerol, J.-Ch. (1997). User Acceptance of Interactive Systems: Lessons from Knowledge-Based and Decision Support Systems. *International Journal on Failures and Lessons Learned in Information Technology Management*. 1(1) (to appear).

Brezillon, P., Gentile, C., Saker, I., & Secron, M. (1997). SART: A system for supporting operators with contextual knowledge. Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97), Federal University of Rio de Janeiro, Brasil (Ed.). pp. 209-222.

Cahour, B. (1993). Modéliser l'utilisateur pour mieux gérer les dialogues. *Cahiers de Linguistique Sociale*.

Clarke, A.A., and Smyth, M.G.G. (1993) A co-operative computer based on the principles of human co-operation. *Int. J. Man-Machine Studies*. 38, pp.3-22.

Damasio, A. R. (1994). *Descartes' error*, Putman's Sons.

Damasio, A.-R., Damasio, H., & Christen, Y. (Eds.) (1996). *Neurobiology of decision-making*, Springer.

Fischer, G. (1990). Communication requirements for cooperative problem solving systems. *Information Systems*. 15(1), pp. 21-36.

Hoc, J.-M. (1996). Supervision et Contrôle de Processus. La cognition en Situation Dynamique, Series Sciences et Technologies de la Connaissance, *Presses Universitaires de Grenoble.*

Karsenty, L., & Brezillon, P. (1995). Cooperative problem solving and explanation, *International Journal of Expert Systems With Applications*. 4, pp. 445-462.

Keen, P.G.W., & Scott Morton, M .S. (1978). *Decision Support System*. Addison Wesley.

Lévine, P., & Pomerol, J.-Ch. (1989). *Systèmes Interactifs d'aide à la décision et systèmes experts*. Hermès, Paris.

Lévine, P., & Pomerol, J.-Ch. (1995). The role of the decision maker in DSSs and representation levels, In J.F. Nunamaker & R.H. Sprague (Eds.), Proceedings of the 28th Annual Hawaii International Conference on System Sciences, Vol. 3, pp. 42-51.

McCarthy, J. (1993). Notes on formalizing context, *Proceedings of the 13th IJCAI*, Vol.1, pp. 555-560.

Newell, A., & Simon, H.A. (1972) *Human Problem Solving*. Prentice-Hall Inc., Englewoods Cliffs, New Jersey.

Sprague, R.H. Jr., & Carlson, E.D. (1982) *Building Effective Decision Support Systems*. Prentice-Hall, New Jersey.

Tversky, A., & Kahneman, D. (1988). Rational choice and the framing of decisions, In Bell D.E., Raiffa H. & Tversky A. (Eds.), *Decision Making*. Cambridge University Press (pp. 167-192).

Winograd, T., & Flores, F. (1986). *Understanding Computers and Cognition: A new foundation for Design*. Ablex.

von Winterfeldt, D. & Edwards, W. (1986). *Decision analysis and behavioral research*. Cambridge University Press.

Woods, D.D., Roth, E.M., & Benett, K. (1990). Explorations in joint human-machine cognitive systems. In Robertson S. Zachary W. & Black J.B. (Eds.) *Cognition, Computing and Cooperation*. Ablex (pp. 123-158).