

## Misuse and nonuse of Knowledge-Based Systems: The past experiences revisited

*P. Brézillon and J.-Ch. Pomerol*

*LAFORIA-IBP, Case 169, University Paris 6*

*4, place Jussieu, 75252 Paris Cedex 05, France*

*Tel.: +33 1 44 27 70 08, Fax: + 33 1 44 27 70 00*

*E-mail: {brezil, pomerol}@laforia.ibp.fr*

### Abstract

It is difficult to determine the number of expert systems or Knowledge-Based Systems (KBSs) that really are operational within companies or administrations. It seems that a large number of such systems have never been used in operations and a rich literature stresses this point. We think that most of these references do not address some dimensions that are of paramount importance. In this paper, we provide a review of the literature according to three neglected dimensions: (1) the differences between automatic KBSs and non-automatic KBSs; (2) the types and importance of decisions that are involved; and (3) the types of data acquisition that is required. Keeping in mind these three dimensions, we review the literature about the acceptance of KBS and their use and point out that most of the observations can be interpreted along these three dimensions. Furthermore, these dimensions permit us to explain some failures and difficulties that have already been pointed out in several scientific domains in which interactivity is crucial, such as Decision Support Systems (DSSs). Our approach brings some new insights on the problems of KBS acceptance and leads us to propose some recommendations.

### Keywords

Expert systems, knowledge-based systems, decision support systems, user-system interaction

## 1 INTRODUCTION

We do not know how many Expert Systems (ESs) or Knowledge-Based Systems (KBSs)<sup>1</sup> that are operational within companies or administrations. Durkin (1993) lists about 2500 ESs, but the exact operational status, as regards operational use, is often not precisely stated. However, some clues exist. For instance, each of the top five Japanese companies has from 20 to 30 operating KBSs (Mizoguchi and Motoda, 1995). Many insurance companies also use such systems (Rowe and Wright, 1993; Meyer et al., 1992).

However, a large number of KBSs has never been used in operations as stressed in the literature. For instance, Majchrzak and Gasser (1991) point out that more than 50% of the systems, which are installed in companies, are not used. The given reasons are: (1) KBSs solved 80% of problems, when users mainly need a support for the 20% others; (2) Integration of KBSs in an organization implies interaction of KBSs with classical software and changes in the organization itself; and (3) KBSs were often imposed on users who know little about the technology.

The above rationales are probably true. However, we think that most of the papers do not address some dimensions that are of paramount importance. Among the neglected dimensions, we want to introduce: (i) the differences between automatic KBSs and nonautomatic KBSs that are intended to be manipulated by end-users; (ii) the types of decision that are involved; and (iii) the types of data acquisition that must be made.

Hereafter, we intend to show that these dimensions explain most of the successes and failures of KBS use as confirmed by the large amount of papers that are already devoted to this topic. We thus provide a revisited view of the literature according to the dimensions presented above. We will moreover show that most of the problems are related to interactivity and that these problems are not specific to KBSs but also exist in any non-automatic Decision Support Systems (DSSs).

We begin, in Section 2, with a presentation of the context of our study. We present the literature revisited in Section 3. On this basis, we discuss then the KBS-development problems in Section 4 and the choice between KBS and DSS in Section 5.

## 2 THE CONTEXT OF OUR STUDY

### 2.1 A basic distinction

Our interest concerns KBSs in operation within companies and administrations. We can split these systems into two classes, namely automatic systems and nonautomatic ones. The difference between them is that an end-user interacts with nonautomatic systems for solving a problem while automatic ones, obviously, run alone. As discussed below, the intervention of the end-user may occur at different steps of the problem-solving process.

---

<sup>1</sup> In the following, we will use ES and KBS interchangeably. The main reason is that the term used now is KBS, but most of the criticisms that are applied to ESs, may also be applied to KBSs. We use the term 'system' to cover both of them when possible.

### *Automatic systems*

Initially, one conceived the building of KBSs as a process of acquiring already 'known' knowledge from experts. Therefore, KBSs mimic the human-expert reasoning to solve decision problems (Hatchuel and Weil, 1992). When integrated in an information system, the KBS automatically acquires its data and provides a problem-solution to the information systems that is assumed to be sound. We call such systems, automatic systems. Process control is the main domain of automatic systems, but one can also find such systems in many industrial processes. They work if the decision is directly dependent on the diagnosis of the current state (Pomerol, 1995). Starting from the experts' knowledge, these systems generally evolve towards refined diagnostic systems in which the model is progressively adjusted by the process engineers. Finally, after refinement, automatic systems act as black boxes whose models may be far from the initial human expertise. For instance, in the SEPT project (Perrot et al., 1993), a temporal cutting out has been introduced to analyze long sequences of recorded signals after incidents to facilitate the diagnosis by the system. This aimed to distinguish incidents very close in time. Such a temporal cutting out does not correspond to the operators' experience in the control command room whose decisions rely on a global analysis.

Automatic systems, which are numerous in some branches, are almost never discussed in the literature and seems to be well-accepted within companies.

### *Nonautomatic systems*

The expert knowledge in nonautomatic systems is also initially acquired from human experts, and users update it while using the system. Being un-integrated, the system, besides expert knowledge, needs specific data defining the situation in which the problem to be solved occurs. Such data are acquired either at the beginning of a session or during the problem solving. While automatic systems directly acquire data from the information system, in many application domains the user must provide the data (e.g., when sophisticated vision is needed). This implies that such systems are nonautomatic. Nonautomatic systems can be classified according to application domains such as: consultancy, clerical work (bank, insurance, etc.), medicine, public information, design.

These nonautomatic systems represent the most important part of KBSs. Very often, it is wrongly assumed in the literature that they represent the totality of KBSs. At least, we can find many papers that are devoted to the failures and, sometimes, the successes of this kind of systems. This is the reason we will especially examine hereafter nonautomatic systems.

## **2.2 Three basic problems**

By definition, a nonautomatic system needs some help from humans. It is mostly important to distinguish between the different natures of human interventions. For simplicity sake, we will only distinguish two types of intervention. The first one concerns data acquisition. The system requires (or prompts) the user to provide some data that are essential for system processing either at the start or during the problem solving. The second type of human intervention concerns what is generally called interactivity. In the context of knowledge systems,

interactivity corresponds to the control of the system by the user to achieve a heuristic search about the decision at hand (Lévine and Pomerol, 1989 and 1995).

### *Data acquisition*

Even when a KBS possesses expert knowledge, it needs the support of the end-user to acquire data on the situation at hand. Most of the data are acquired at the beginning of the session, but also during the system reasoning. When other data are needed, the system generally prompts the user. The main obstacles to automatic data acquisition are met when performing tasks in which natural language or reading of manuscript or 'good enough' vision capabilities are needed.

In all these three cases, the end-user plays the role of a data gatherer, who need not know why the system needs a specific data, but is only required to provide them. This may result in sequences of questions that appear incoherent for users (Keravnou and Washbrook, 1989; Woods et al., 1990). This is particularly frustrating for end-users and may lead them to reject the system.

Also, the user is very sensitive to the quantity (in term of reduced workload or usefulness) of outputs with respect to the time and the work that is necessary to enter the data. We can summarize the situation by considering the symbolic ratio R:

$$R = \frac{\text{Quantity of outputs}}{\text{Workload necessary to provide data}}$$

A system in which  $R < 1$  has very little chance to be accepted, and eventually used. This situation is very frequent in clerical work where users are (still!) either obliged to bridge the gap between natural language or hand writing and the machine. Note that the situation is not symmetrical because once the data are entered into the machine, the system is perfectly able to produce, on its own, various outputs (e.g., a letter, a report or a record). As a consequence, one can improve R by increasing the numerator, which in turn, requests a serious integration of the system within the information system. Also, the time spent during a consultation with a KBS is a determining factor of its acceptance (Pomerol and Retour, 1990). Roughly speaking, this time corresponds to the time spent for data acquisition. Moreover, the end-users may have other sources of information (control boards, telephones, etc.) and wish to direct the machine's attention to different subsets of data, different domain issues, and different hypotheses to help the KBS to refine its understanding of the problem. Generally, the system is not able to accept such additional data.

### *What type of decision?*

The result of the system running is either a diagnosis, a proposal of action or a mere suggestion. In order to anticipate the end-user's reactions, it is absolutely necessary to understand the nature of the intended output of the system. When the expected output is a diagnosis, end-users receive the diagnosis either as a confirmation or as a disagreement of their own diagnosis. In the latter case, the user and the system may enter into a dialogue to solve the conflict. (It is also the case in training.) This is well documented in the literature and

leads to questions about explanation, understanding of the output, etc. For instance, several authors mention that users' questions in traditional KBSs do not reflect the end-users' actual needs, e.g., see (Kidd, 1985; Gilbert, 1987). The main problem is that the system reasoning may be quite different from the user's reasoning. Thus, users face both the problem they seek to solve and the logic of the system at the same time. Moreover, the system may not solve the exact problem that the user actually wants to solve or needs help in solving it. For example, present diagnostic systems perform classification procedures to answer the questions: What is the fault? Or What is the remedy? Empirical studies have shown that users, engaged in a range of diagnostic tasks, rarely ask experts either of these questions (Kidd and Sharpe, 1988). More often, to solve their problem, they want answers to questions such as: Why did fault X happen? Will remedy Y cure it? Can I test W without affecting the level of Z? Another point is that users do not want abstracted information, only information in the specific context of their particular problem.

Consider now the case in which the system output is a decision. There are a number of examples in management and consulting: investment choice, different types of underwriting, responsibility assignment, facility location, etc. Three prominent attributes characterize these decisions in an opposite way to control process decision: (1) their consequences are difficult to assess because the results are delayed (we discuss this point later); (2) the results are important in term of return (for example, decide whether or not to build a new facility); and (3) the decision is irreversible (e.g., once the building of a factory has begun, there is no way to stop or modify the process except for details). What is surprising in such contexts is that the system designers may have thought that the system should make the decision alone. These types of decision are far too risky and important to be left to a system. This is the realm of the Decision Support Systems (DSSs). The designer must understand that, in this case, what users need is a help and further information and not an automatic decision making. Thus, we are in a situation in which the user needs to understand the models that are used by the system and have to be confident of the system outputs. Among the prominent questions bothering the decision makers is the link between the possible actions and the results. The answer to this question depends on what will happen in the future. This is the look-ahead problem (Pomerol, 1995). This problem, together with subject preferences, deserves a special subsection.

### *Incidence of the future and of preferences*

When coping with a decision problem, the decision maker faces two complex issues: (i) How to anticipate the results of a given action?; and (ii) How to choose among the possible actions taking into account their uncertain consequences? In the second question, the choice is given by a translation of the decision maker's preferences onto the set of uncertain issues. However, most often, systems address neither of these two issues. It is therefore not surprising that people confronting real decisions do not believe in ES or KBS support.

In most systems, one implicitly supposes that there is a direct link between the current state of nature, the actions and the issues. In other words, the system runs in a certain world and future does not matter (Pomerol, 1995). It is the case in many industrial processes where the time interval between two actions is so narrow that the uncertainty of the environment cannot entail important consequences. We thus discover again one more reason for the success of KBSs in the control process or industrial reactive systems.

The other implicit assumptions generally made by KBS designers is that everybody has the same preferences. Again, it is generally true in industrial processes: one wants to maximize the output of the process. It is also the case in medicine (and more generally in diagnosis) where one wants to cure the system or the patient even if people may disagree on the tradeoff between the expected result and the necessary resources for the tests. In many other cases (e.g., investment choice, location, etc.), it is unlikely that experts, users, or modelers have the same preferences. The decision-maker (or end-user) requirement of introducing his own preferences often surfaces in needs such that those expressed by Kidd and Sharpe (1988):

- Set out his own constraints on an acceptable solution, e.g., 'It must be quick' or 'I can't take the back panel off because I haven't got the tools';
- Put forward his own plans, solutions or explanations for evaluation, e.g., 'Will swapping that component clear it for good?' or 'Did it fail to work because of the sequence I used?';
- Reject or request alternatives to solutions proposed by the expert, e.g., 'I've already tried that and it didn't work' or 'Are there any cheaper options?'.

It is noteworthy that the fundamental requirement of system users of being not deprived of expressing their preferences is generally recognized in DSSs. We have argued that it is one, among the best virtues of DSSs, to permit this expression through heuristic search (Lévine and Pomerol, 1989 and 1995).

According to the three main difficulties of (1) data acquisition, (2) type and importance of a decision, and (3) the role of the uncertainty and of preferences, it is interesting to review the literature to see if our distinctions are relevant and explain, at least partly, the lack of satisfaction of KBS users.

### 3 THE LITERATURE REVISITED

Not surprisingly, most of the papers focus on the relationships between the system and the end-user. This confirms that automatic systems do not raise many problems. When data acquisition is possible via the information system, and when actions are sufficiently 'continuous' so that the consequences are almost certain, automatic systems acting as sophisticated automata rise no other problem than a good modeling and a good developing team. The literature being almost exclusively devoted to nonautomatic systems, we restrict ourselves to these systems in this section.

#### 3.1 The relationships between the end-user and the system

There are several common arguments about the misuse and nonuse of KBSs. We can classify most of these arguments by using the attributes that we have introduced previously. Many comments are related to data acquisition. It appears that users do not want to be used as data gatherers. Remember that KBSs have been designed initially like oracles for helping users--considered as novices--in their tasks (Karsenty and Brézillon, 1995): all the knowledge is within the machine. The noble part of the task (the reasoning) being in the system, the role of the user is to:

- act as an interface between the machine and its environment,

- be a natural language interpreter, the eyes and hands of the machine,
- function as a passive data gatherer for the machine (a data entry clerk (Fisher, 1990)).

Now, it is obvious that no one wants to use a system during a long time if it is for playing the above role. The system then suffers of this irremediable flaw, and is quickly rejected, except for provisional use for training.

It is also mentioned that KBSs are not able to take into account the fact that users become increasingly experienced. This may also be interpreted as a desire to alleviate the data-acquisition burden for experienced users. Very often, one can observe that users cease to use a system because they already have forged the decision in their mind before the end of the data acquisition.

Leaving the data acquisition aspect, we found references in which authors put forward many observations relative to users' wishes. Users would like to have the possibility to: generate partial solutions by themselves; try to explain the problem and identify a solution; provide spontaneously information; come back on the data provided previously; ask their questions; and give their answers in their language. The flexibility requirements involve an interruption device and the expression of a partial explanation. When users have troubles with the system reasoning, they must infer machine intentions, resolve impasses and recover from errors (person or machine) that led the machine expert offtrack, and have a limited set of possible interpretations. One reason is that users generally must follow underspecified instructions, and the unique way to interrupt the system reasoning is to abort it.

The main attempt to convince users was to provide them with explanations. However, it was a pitfall because the explanation was only **from** the system to the user, the user does not intervene in the building of the explanation (Karsenty and Brézillon, 1995). Rejection of an explanation may be due to a misunderstanding of the explanation, or the need of another type of explanation, or the inability to use the explanation. Clearly, there is a need for alternative explanations, and a reactive approach to explanation is not sufficient (Moore and Swartout, 1990).

The following step was to tailor explanation to users' needs. Among the various interesting attempts, there are: (i) Adapting the degree of details to the users' knowledge (Wallis & Shortliffe, 1984; Swartout, 1983); (ii) Adapting the explanation types to the users' knowledge through a user model (Paris, 1990; Moore and Swartout, 1990); (iii) Adapting the explanation types to the users' goals (McKeown et al., 1985; Van Beek, 1987). In most of these attempts, explanations are conceived as texts that have to be generated, again from the system to the user.

Most of these common arguments deal with the mostly important capacity of the system to allow end-users exploration, which is at the root of their 'what-if' implicit analysis and of the look-ahead. As already mentioned, one of the main concern of the decision maker is to bridge the gap between the actions and the consequences. The user needs a friendly help to perform the look-ahead that is a crucial movement of decision making. This help includes the understanding of the system reasoning and the need of exploration and explanation capabilities among which we can find in the literature:

- Provide system solutions with concrete examples and convincing arguments;
- Solve problems in a comprehensible way for the user, mainly in the user's semantics, not only in terms of the official norm;
- Provide explanations in the user's terms when no solution is found;
- Offer training functions;

- Allow users to return easily to previous states of their information search.

All these users' requirements for a better control of the system reasoning are more or less similarly expressed in DSS literature. They seem to be very common in any interactive system in which users want to impose their preferences via the control of the exploration. For important decisions discussed in Section 2, the final choice depends on users' preferences, and users try by a number of ways to recover the system control, even to abort it if necessary.

In the context of 'what-if' analysis, the need for approximated data and for a robustness evaluation of the system recommendation is a classical requirement of decision aid in DSS framework. This requirement is very frequent and leads to questions as: How to deal with approximated solutions (e.g., the range of prices is around ...)? Curiously, the question of approximate data is often tackled in KBSs by various means, such as fuzzy logic. However, it is rarely mentioned as a user's requirement. Conversely, the literature on DSSs often points out this question.

The most convincing evidence that users will not give up their right to control a decision is that many rejected systems have been revamped into a training system. The user may deviate from the original goal of the system either for a subuse (e.g., training or 'what-if' question) or another goal (e.g., note ideas in an electronic agenda that becomes a record of the user's history). Not only users may deviate the system function, but also knowledge engineers: The most popular case is those of MYCIN becoming a series of tutoring systems (Clancey, 1986).

Using a system for training acknowledges that the system is valuable within a given context (preferences, implicit assumptions, contextual assumptions, etc.) but, in the practical situation at hand, many extra adaptations and interpretations remain necessary. Users consider therefore that the system is unable to make the decision and that they are unable (or do not want to) to tailor the system at their needs. Thus, the system is valuable for training in what is considered as a theoretical framework, but the reality, which requires more subtle insights which remain out of the system's grasp. Another definitive argument is that either the system is able to make the decision and users disappear (automatic system) or the system is unusable. Then, we enter into the DSS realm and KBS users' claim joins the usual requirements of DSS users. Designers must acknowledge that there is no alternative way between an automatic system and human decision. In the latter case, the system is either a training system or a DSS, **not a knowledge-based decision matter**.

### **3.2 The System and the Organization**

Many authors have considered KBSs from an organizational viewpoint and have attributed nonuses or misuses of KBSs to organizational failures. For example, see references in Pomeroy (1990). The common statement is that the organization fails to integrate the system. Changes in work is the first cause of conflict when introducing into organization new information technology systems (Agro et al., 1995). The authors often refer to the difficulty of introducing a new technology into an organization that should accept changes (Hatchuel and Weil, 1992). Most organizations prefer to take a low risk position when considering a new technology. As such, projects that require the minimum resources and have the maximum likelihood of success are preferred (Durkin, 1993).

Typical organizational implementation problems include inadequate preplanning for the social as well technology changes concomitant with modernization efforts. Here are some



causes of failures that one can find in the literature: inadequate training, workteams not supported by management, failure to anticipate organizational resistance, incongruent pay systems, failure to understand how work is really made, increasing of the workload, etc. Instead of simply modifying some work procedures, entire units may need to be reorganized and appraised in different ways, new cultural behaviors may be to be instilled, and every one in the organization may need to be reskilled (Majchrzak and Gasser, 1991). Another problem in a working environment may concern the possible isolation of users working with KBSs among their colleagues that pursue their work in the usual way.

The problem raised by data acquisition when integrating the system in an information system (see above) is sometimes interpreted as an organizational or software problem. Stand-alone KBSs would move towards systems embedded in other conventional software programs (Durkin, 1993). Commercial tools are black boxes and thus are of little use to companies. Developers have to customize KBSs and to integrate them in existing information systems. Whereas they often prefer to establish their own expert-system building methodologies and think to integrate when the system is completed.

For a large part, the observations about organizational changes are not different from the usual views about introducing changes within organizations and go back to Lewin (1947, quoted by Pomerol (1990)). Any introduction of a new system into an organization either MISs (Management Information Systems), DSSs, EISs (Executive Information Systems), etc., rises the same types of questions. Chandrasekaran (1994) gives some interesting comments on such "smart" systems.

Other views are related to knowledge objectivity in KBSs. Among them, the risk of a possible direct control of the users' work by their chief (Hatchuel and Weil, 1992); a fear of a comparison of the KBS capabilities relatively to the present user; the eventual finding of user's mistakes by the KBS (Ackerman et al., 1992); and the users' fear to be in danger of losing their job (Sakagushi et al., 1987). The transfer of expert knowledge into a system also may be transfer of power (Belanger et al., 1995). For instance, KBSs, which are never tired, can have higher productivity than users. Users may consider to be replaced by KBSs as a danger for them, while experts maintain their power because of their skills and access to special information that is valued by high individual ranking.

However, KBSs may be a way for a top-down transfer of competence and responsibilities in normal condition and, conversely, a bottom-up transfer of competence in critical situation (de Terssac and Chabaud, 1992). This move of responsibility among actors is generally well perceived: users' work is favoured, chiefs are relieved of a part of their routine work (Pomerol, 1990; de Terssac and Chabaud, 1992).

In our opinion, it appears that the users' level of competence does not depend on the intrinsic characteristics of the used technology (a KBS is a tool among other tools for users), but on the specifications and decomposition of the work, and on the modes along which the work is structured in a company. We think that the introduction of KBSs in organization is a step of the computerization process that stretches now from structured to less structured tasks. This modernization may last for many years accompanying the movement of information computerization.

Besides the organizational obstacles related to change, one of the main impediments on the way of KBS integration is data acquisition. In many clerical tasks, people handle manuscript letters, listen people talking natural language, etc. The computer technology does not allow a direct transfer of this information into the machine. Thus, a human operator is

necessary to introduce the data. On the one hand, this is the main reason for the relative failure of introducing KBSs in many administrative tasks. When automatic data acquisition is technically impossible, the unique way to integrate the system is to invent work procedures that oblige the end-user to effectively use the system. On the other hand, many systems have been designed to tackle decisions in which the human component is mostly important, e.g., combining human competences (Hatchuel and Weil, 1992) or of the type discussed in 2.2.2. In these cases, it is obviously not surprising that the system may be rejected, and, in the best case, becomes a training system.

For instance, operators in a control room argue that (Huguet et al., 1995):

- they want to avoid the system's control which could be used to check the amount of time spent on each order;
- they intend to protect themselves against a possible load increase as the system will allocate production orders as soon as the resources become available;
- they wish to avoid an additional management task which is little compatible with the production work;
- they have their own doubts about the advantages of the system being fed with data in real time.

Part of experts' knowledge cannot be codified (Hatchuel and Weil, 1992). This concerns the use of the know-how in communicational contexts where predominates interaction among several actors. Such a meta-know-how is generally described in nonwritten rules. A non-written rule takes into account the real context of the work at a given moment. Such a contextualization of procedures permits the solution to be tailored to the conditions of the task; such "makeshift repairs" permit the executing actors to reach the efficiency that is wished by those that specify the work to do. It is a manner to reach the solution whatever is the path that is followed. The validation of non-written rules is more linked to the result than the procedure to reach it, a logic of the efficiency (de Terssac, 1992). Such knowledge is for a personal use, not to be used by others.

## **4 KBS-DEVELOPMENT PROBLEMS**

### **4.1 The different actors**

Several human actors intervene during the lifecycle of a KBS: the manager who decides the construction of the KBS, the expert whose knowledge and heuristics have to be captured by the system, the knowledge engineer who models the expertise, the end-user who gains practical experience by frequently using the KBS, and the occasional end-users (novices, users from other departments that occasionally consult the KBS, etc.). The importance of each human actor varies according to the phase of the KBS design. Some of them only intervene during a small number of phases. For instance, it is frequent that the main user does not intervene in the early phases of the design, and this is generally a problem for the acceptance of the KBS later. This probably implies that users intervene in the KBS design, whereas they are confronted with highly nontransparent systems that are far from easy to use (Oberquellé et al., 1983).

After the system development, some actors are authorized to modify the knowledge base. The authorized actors generally are: the domain expert, the knowledge engineer, and

eventually the main user. The user that has to make strategic decisions, is sometimes an authorized actor. Conversely, a user who either follows operating procedures or supervises the process is not an authorized actor. The updating task of the knowledge base is crucial because a KBS that is not maintained is a dead one. When users must intervene in the knowledge base for maintenance, they generally prefer to add rules rather than intervene directly on the structure of the KBS itself. This leads to a complex structure of the KBS that becomes quite unexplainable for other humans (e.g., experts) and for themselves after a while. This situation is particularly crucial when KBSs are developed in an empirical way without any written trace. Maintenance becomes a fastidious task for the user, and added to the task at hand. We also distinguish actors and roles played by actors. Actually, a role is not always held by the same human actor. For instance, the design may begin with a domain expert and finish with another one. The change of main user, for instance, implies a need for training at any phase of the KBS lifecycle. As a consequence, a part of the experience-based knowledge may be lost (e.g., information in the way a KBS is coded). Moreover, a given human actor may play different roles too. For instance, main user and maintainer may be considered as two roles that are played by the same actor. This depends on the relationships between the user and the system in the achievement of the task at hand. The system may play different roles too: tutor, suggestor, advisor, critic.

## 4.2 Knowledge engineering

*"Knowledge engineers are like priests; They receive the "Word" from experts above, add nothing to the content, but codify it accurately into written rules, and pass it down to ordinary folks as commandments to live by" (Clancey, 1993).*

A missing link is that experts provide knowledge in the context of a given situation. The contextual dimension of the acquired knowledge is generally not taken into account, whereas experts--as, for example, decision makers--heavily rely on information coming from the environment, often in an intuitive or unstructured way (Pomerol, 1993). The decision makers try to identify the current state by reference to their past experience. This means that we can assume that the subject has recorded many situations or states he has already met or has learnt about (Pomerol, 1995). Thus, the humans' expertise must be considered in the context of their action, and that context must be acquired.

The knowledge acquisition comprises various biases due to the interpretation and the way to code knowledge. The Knowledge Engineer (KE)'s interpretation of the experts' knowledge is made in the KE's context, not necessarily in the expert's one, and, moreover, the operating context may differ of the the two previous ones. The KE's context relies on the representation formalism that has generally been chosen for reason others than for the knowledge acquisition (e.g., for using a software already existing in the company). Expert's knowledge is reconstructed by the KE and knowledge acquisition can be analyzed as a process of knowledge creation (Cooke, 1994). When the human expert provides knowledge for a given context of problem solving, this context is generally not totally acquired by the KE.

Moreover, KEs introduce additional pieces of knowledge that obscure the knowledge bases. For example, the rule ordering that has been shown to directly control the firing of rules and may produce incorrect results in the system reasoning. In a system such as GUIDON (Clancey, 1986), screening clauses are added to the rule base to limit the effects of

triggered rules. The same conclusion has been reached in the SEPT project (Brézillon et al., 1988). Screening clauses obstruct the facility to change the knowledge base due to the continuous changes required by the domain expert. As a consequence, such a control on the system reasoning, which generally is ignored by experts, implies that experts cannot always follow and validate the system reasoning.

KEs generally have a false understanding of users' needs. When users express their needs in terms of **what** they want, the KE treats the request in terms of **how** to do. This is the well-known problem of the logic of use against of the logic of functioning that leads to a conflict between what the system does and what users' needs are. Users need information that is tailored to **their** psychology **not** to that of KEs (Wood and Wood-Harper, 1993). This leads to a difficulty for implementing a KBS in the organizations and to nonadaptability and misunderstanding during the users' training.

Also, large companies design and develop by themselves KBSs (Mizoguchi and Motoda, 1995). This insures a direct compatibility with existing software in the industry. It is generally a computer engineer that plays the role of the KE (and it is a reason to distinguish actors and roles, mainly because experts often develop KBSs by themselves).

### **4.3 Partial conclusion**

One way to improve the relationships between the user and a KBS is to permit the user to have an active role, be alive to the interest of the KBS and contribute to its coming. However, we must avoid the optimistic view that user involvement is sufficient enough to insure the success of a project (Agro et al., 1995). The user wants to follow easily the evolution of the system reasoning and communicate with the KBS in an accessible language. On the other hand, the KBS must minimize user memory overload, provide feedback (let users know what effect their actions have on the system), provide clearly marked exits and shortcuts, help users ask questions and give an overview of the organization (structure) of the information. The KBS also must take into account: the components forgotten by the user; the components added by the user; the parameter values often used by the user; the ordering constraints not respected by the user (Levrat and Thomas, 1993).

If the target is an operational use of a system, it is unlikely to reach it out of an end-user centered approach. The first step of this approach is a careful analysis of the user's task without, and with the system in the user's working environment. Indeed, many KBSs addressed mythological tasks which are neither interesting for the user nor for the organization. A typical example is given by many ESs intended to help salesmen during commercial transactions. Nobody wondered how, during the fragile dialogue between a buyer and a vendor, the latter could manage an interaction with the computer and on what topic.

Besides the analysis task and its working context, including data acquisition, the designer must understand the issues at stake. Many working situations ask for humanity rather than for expertise (e.g., communication, intention, commitment). If the human component at stake is important, it is unlikely that a system be trusted. The designer must, from scratch, be concentrated on helping the decision maker rather than try to replace him by the system. This is a typical of DSS framework (Keen and Scot Morton, 1978). The same conclusion can be reached when the future or the preferences really strongly matter. One cannot hope either to model preferences of each user or to capture all the subjective probability, this suffices to impede many systems to be used. Designers must reduce their ambition to only model the part

of the knowledge which is common to all the users. This implies that the system be highly interactive so that each user can introduce his preferences and subjective anticipations.

## 5 KBS or DSS?

The main criticisms presented in the previous sections concern the lack of interest for the users, the poor exploration capabilities and the type of work (or decision) that the system is intended to do. To overcome these difficulties, there are only two solutions: integrate and automate, or if automation is not possible or wished, to be more aware of users' needs. In the latter case, the user must be able to intervene at different steps of the system development. Moreover, the user would intervene at various levels: in the design loop, in the problem solving, and in the knowledge base. Moreover social issues like the introduction of the KBS in a company must be prepared.

The designers must anticipate the integration of the user (if any) in the information system, including KBS modules. Furthermore, one must plan the exact role of the user, either data gatherer or interacting actor. In the latter case, the system must help the user to have an active role (Carr, 1992; Woods and Roth, 1988). Indeed, the task must be accomplished interactively by the two interested persons who are the KBS and the user. The active role implies that there is a shared control of: the interaction (e.g., the user may interrupt the system without aborting it), the knowledge base (e.g., to provide spontaneously information or for maintenance purpose), and the problem solving.

The shared control of the problem solving implies a decomposition of the task, some tasks being jointly solved by the system and the user (de Greef and Breuker, 1989; 1992). Indeed, a shared control in the noble part of the task (i.e., the data treatment) may permit to exploit the complementarity of the competences of the user and the system. This is particularly important when alternative solutions appear during the problem solving.

All the recommendations described in this section leads to consider the system and the user as a 'joint cognitive system' (Woods et al., 1990). These views about joint cognitive system are very close to those put forward for years in DSS literature (e.g., Keen and Scott Morton, 1978; Lévine and Pomerol, 1989). We think however that we must go a step further. Indeed the system must be an intelligent assistant system (Boy, 1991; Brézillon and Cases, 1995). For this, we consider three aspects:

- (1) Explanation. The role of the explanations in a cooperative problem solving must be revised, because explanation is intrinsic to any cooperation (and, conversely, the system and the user must cooperate to build an explanation) (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). Note that we have here a kind of explanation from the user to the system, when actually one only considers explanation from the system to the user.
- (3) Context. It is necessary to make the context of the user-system interaction explicit in the problem solving to provide relevant explanation and acquire incrementally knowledge (Brézillon and Abu-Hakima, 1995).

All these views simply generalize some common aspects discussed in the DSS community since many years. Apart to stress some topics about knowledge modeling, it now

appears that KBSs do not differ from DSSs and encounter the same problems of man-machine integration.

## 6 CONCLUSION

There is an abundant literature about misuse and nonuse of KBSs. It is mostly important to turn back to this experience and to understand what are the real problems with such systems.

Our analysis shows that most of the failures occur with nonautomatic systems and are related to data acquisition and interactivity weaknesses. The first common pitfall is to confuse data acquisition and interactivity. Users refuse the work overload of introducing data, except if they are convinced that the output really worth it. As a consequence, in domains where data are issued from natural language exchanges or manuscript writing, up to now, it is unlikely to oblige end-users to act as a data provider in front of any computer system, KBSs included.

The second question, which arises similarly in DSSs and other advanced information systems, is the problem of interactivity. It is very difficult to define what interactivity means, and to handle the user integration into an information system. This raises many issues as explanation, contextual information, learning. The main result in this domain is that the user wants to be helped or complemented by the system, but not replaced (Keen and Scot Morton, 1978). It is therefore important to analyze the task and the type of decision that the user is facing. The most important point being to recognize that some decisions are intrinsically human in nature and that the destiny of the system is not to make the decision.

The third point is that organizational problems about changes have already been described with the introduction of many advanced information systems. Again, DSSs provide good examples and guidelines.

Finally, the increasing complexity of systems, devices and organization entails the need for sophisticated training of people. KBSs have an important role as a part of training systems. This, in turn, raises new issues about explanation, trainee's experience, contextual changes. All these problems enter into the general setting of man-system cooperation and joint cognitive systems.

It is more and more difficult to distinguish between DSSs and KBSs. Most of the views on users' involvement during design and development (mainly KBSs) and progressive development (DSSs), understanding of the reasoning (KBSs), confidence and acceptance of the model (DSSs), explanation facilities (KBSs), enhancement of the dialogues capabilities (DSSs), all of these views converge towards some common knowledge related to the design of advanced cooperative information systems. This knowledge focuses on three main issues, namely, the user involvement, the careful analysis of the task at the early steps of the design, and the integration of the system in an organization. The design of operational nonautomatic systems requires the control and the adequate answers to these three challenges.

## 7 REFERENCES

Abu-Hakima, S. and Brézillon, P. (1994) Knowledge acquisition and explanation for diagnosis in context, *Research Report 94/11*, LAFORIA, University Paris VI, Paris, France.

- Ackermann, W., Giustina, L.D., Gremion, C., Gremion, S. and Pomerol, J.Ch. (1992) Etude sur les nouvelles technologies et l'aide à la décision, *Rapport de ENA Recherche*.
- Agro, C., Cornet A. and Pichault F. (1995) L'implication des utilisateurs dans les projets informatiques : un scénario en quête d'auteurs, *Gérer et Comprendre*, **41**, 33-44.
- Belanger, F., Burns, M.B. and Will, R.P. (1995) Transference of power: Shifting expertise to expert system developers, *Proc. of the 11th Conf. on Artificial Intelligence for Applications*, 65-71.
- Boy, G. (1991) Intelligent Assistant Systems, Academic Press, London, Knowledge-Based Systems, **6**.
- Brézillon, P., Fauquembergue, P. and Hertz, A. (1988) SEPT, an expert system approach for the monitoring of EVH substation control equipment, *Proc. of the Symposium on Expert Systems Application to Power System*, Stockholm-Helsinki, 1988, 6.9-6.13.
- Brézillon, P. and Abu-Hakima, S. (1995) Using knowledge in its context: Report on the IJCAI-93 Workshop, *The AI Magazine*, **16**(1), 87-91.
- Brézillon, P. and Cases, E. (1995) Cooperating for assisting intelligently operators, *Proc. of the International Workshop on the Design of Cooperative Systems*, INRIA ed., 370-384.
- Carr, C. (1992) Performance support systems: A new horizon for expert systems, *AI Expert*, 44-49.
- Chandrasekaran B. (1994) AI, knowledge, and the quest for smart systems, *IEEE Expert*, December, 2-5.
- Clancey, W.J. (1986) From GUIDON to NEOMYCIN and HERACLES in twenty short lessons: ORN final report 1979-1985, *The AI Magazine*, 40-60.
- Clancey, W.J. (1993) Notes on "Epistemology of a rule-based expert system", *Artificial Intelligence Journal*, **59**, 197-204.
- Cooke, N.J. (1994) Varieties of knowledge elicitation techniques, *Int. J. Human-Computer Studies*, **41**, 801-849.
- Courbon, J. Cl. (1992) Expert systems as a methodological step in conventional decision support system, *Economics and Cognitive Science*, P. Bourguine and B. Walliser (Eds.), Pergamon Press, 383-386.
- de Greef, H.P. and Breuker, J.A. (1989) A methodology for analyzing Modalities of System/User Cooperation for KBS, *Proc. of the 3rd European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Paris, 462-473.
- de Greef, H.P. and Breuker, J.A. (1992) Analysing system-user cooperation in KADS, *Knowledge Acquisition*, **4**, 89-108.
- Durkin, J. (1993) Expert Systems. Catalog of Applications, Intelligent Computer Systems Inc., PO Box 4117, Akron, Ohio 44321-117.
- 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.
- Hatchuel, A. and Weil, B. (1992) L'Expert et le Système, *Economica*, Paris, France.
- Huguet, M.J., Erschler, J., de Terssac, J. and Lompré, N. (1995) Negotiation based on constraints in cooperation, *Proc. of the International Workshop on the Design of Cooperative Systems*, INRIA ed., 109-126.
- Karsenty, L. and Brézillon, P. (1995) Cooperative problem solving and explanation, *Int. J. on Expert Systems With Applications*, **8**(4), 445-462.

- Keen, P.G.W. and Scott Morton M.S. (1978) Decision support systems, Addison-Wesley Publishing Company, Readings, MA.
- Keravnou, E.T. and 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.
- Kidd, A.L. and Sharpe, W.P. (1988) Goals for expert systems research: an analysis of tasks and domains, In: *Annual Technical Conference of the British Computer Society*, Cambridge University Press, 146-152.
- Lévine, P. and Pomerol, J.Ch. (1989) SIAD et systèmes experts, *Hermès*, Paris.
- Lévine, P. and Pomerol, J.Ch. (1995) The role of the decision maker in DSSs and representation levels, *Proc. of the 28th Annual Hawai International Conference on System Sciences*, Nunamaker J.F. & Sprague R.H. (eds.), IEEE Computer Society Press, Los Alamitos, USA, 3, 42-51.
- Levrat, B. and Thomas, I. (1993) Tailoring explanations to the user's expectations: a way to be relevant, *IJCAI Workshop on Explanation and Problem Solving*, 1-10.
- McKeown, K.R., Wish, M. and Matthews, K. (1985) Tailoring explanations for the user, *Proc. of IJCAI'85*, 794-798.
- Majchrzak, and Gasser, L. (1991) On using Artificial Intelligence to integrate the design of organizational and process change in US manufacturing, *AI and Society Journal*, 5, 321-338.
- Meyer, M.H., Detore A., Siegel S.F. and Curley K.F. (1992) The strategic use of expert systems for risk management in the Insurance Industry, *Expert Systems With Applications*, 5, 15-24.
- Mizoguchi, R. and Motoda, H. (1995) Expert systems research in Japan, *IEEE Expert Magazine*, 14-23.
- Moore, J.D. and 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.
- Oberquelle, H., Kupka, I. and Maass, S. (1983) A view of human-machine communication and co-operation, *Int. J. Man-Machine Studies*, 19, 309-333.
- 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.
- Perrot, L., Brézillon, P. and Fauquembergue, P. (1993) Towards automatic generation of knowledge bases for diagnosis systems in the field of power systems, *ESAP'93*.
- Pomerol, J.Ch. (1990) *Systèmes experts et SIAD : enjeux et conséquences pour les organisations*, T.I.S., 3(1), 37-64.
- Pomerol, J. Ch. and Retour, D. (1990) L'introduction et l'utilisation des systèmes experts dans le tertiaire, *Bilan d'Etudes*, C3E, 10 rue Saint Claude, 75003 Paris.
- Pomerol, J.C. (1993) Multicriteria DSSs: State of the art and problems, *Central European Journal for Operations Research and Economics*, 2(3): 197-211.
- Pomerol, J.C. (1995) Artificial intelligence and human decision making, *Proc. of Euro XIV*, Jerusalem, R. Slowinski Ed., 169-196.



- Rowe, G. and Wright, G. (1993) Expert systems in Insurance: A review and analysis, *Intelligent Systems in Accounting, Finance and Management*, 2, 129-145.
- Sakaguchi, T., Tanaka, H., Uenishi, K., Gotoh, T. and Sekine, Y. (1987) Prospects of expert systems in power system operation, *9th Power Systems Computation Conference*, Cascais, Portugal.
- Swartout, W.R. (1983) XPLAIN: a system for creating and explaining expert consulting programs, *Artificial Intelligence*, 21(3), 285-325.
- Terssac (de), G. (1992) "Autonomie dans le travail", *Série Sociologie d'Aujourd'hui*, Presses Universitaires de France.
- Terssac (de), G. and Chabaud, C. (1992) Impact social des systèmes experts. Repères pour une méthodologie de conception, *Technologies, Idéologies, Pratiques*, X(2-4), 7-28.
- Van Beek, P. (1987) A model for generating better explanations, *Proc. of the 25th Conference of the Association for Computational Linguistics*, Stanford, 215-220.
- Wallis, J.W. and 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.
- Woods, D.D. and 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. and Benett, K. (1990) Explorations in joint human-machine cognitive systems, In: *Cognition, Computing and Cooperation*, Robertson S, Zachary W & Black JB (Eds.), 123-158.
- Wood, J.R.G. and Wood-Harper, A.T. (1993) Information technology in support of individual decision-making, *J. of Information Systems*, 3, 85-101.

## 8 BIOGRAPHY

**Patrick Brézillon** is a researcher at the National Center of Scientific Research in France (CNRS). In 1983, he received his These d'Etat in natural sciences at the Pierre and Marie Curie University. The topic of his thesis concerned the mathematical modeling of the calcium metabolism as a self-oscillating nonlinear model. Now, he aims to merge the mathematical modeling with the AI approach and is concerned with topics such as cooperation, context, explanation and incremental knowledge acquisition in the framework of 'intelligent' cooperative systems.

**Jean-Charles Pomerol** is Professor of Computer Science and the head of the LAFORIA-IBP, a laboratory at University Paris 6 associated with CNRS. This laboratory is entirely devoted to Artificial Intelligence and Speech Recognition. Jean-Charles Pomerol defended his These d'Etat in convex analysis in 1980. Then, he turns to decision theory and decision support systems. His current interests are about the design and development of 'intelligent' decision support systems. He is the author or co-author of many papers and four books, concerning expert systems, decision support systems and multicriteria decision.