A version of this paper has been published in: *International Journal of Human-Computer Interaction*, 10 (3), pp. 193-231, 1998.

A Review of "intelligent" human-machine interfaces in the light of the ARCH model

Christophe Kolski
Emmanuelle Le Strugeon

Laboratoire d'Automatique et de Mécanique Industrielles et Humaines (LAMIH) [1]

An important field of current research is that of the design and implementation of so-called “intelligent” human-machine interfaces (HMI). This article reviews a number of concepts concerning so-called “intelligent” interfaces, taking as its starting point the well-known ARCH model of HMI, with particular concern for the ability of such interfaces to be flexible, adaptive, tolerant of human error, and supportive both of human operators and intelligent agents.

1. INTRODUCTION

A major role of HMIs is to bridge the gaps which exist between humans and machines (Card, 1989). In this perspective, research on so-called “intelligent” interfaces appeared at the beginning of the 1980s. A common definition of an intelligent interface is one which provides tools to help minimize the cognitive distance between the mental model which the user has of the task, and the way in which the task is presented to the user by the computer when the task is performed (Hancock and Chignell (1989)). According to Chignell et al. (1989), an “intelligent” HMI is an “intelligent” entity which mediates between two or more interactive agents, each of which has either imperfect understanding of the way in which the others act, or an imperfect understanding of the way in which the others communicate. Figure 1 shows the obvious importance given both to the task and the user in an early representation of an intelligent interface, in the well-known article by Chignell and Hancock (1988).

---

[1] Correspondance should be sent to Prof. Christophe Kolski, Laboratoire d'Automatique et de Mécanique Industrielles et Humaines, LAMIH - URA CNRS 1775, Le Mont Houy, BP 311, 59304 Valenciennes cedex, FRANCE
e-mail: kolski@univ-valenciennes.fr

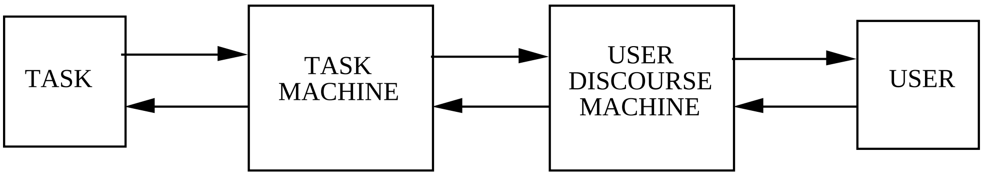

Figure 1. An abstract representation of an intelligent interface (from Chignell and Hancock, 1988)

This research domain originated in the field of Artificial Intelligence, cognitive science, and computer science in connection with models of tasks, human behaviour, user models, reduction of human error, etc., and a brief review of some of this background is given in Part 2 of this paper. [2]

Since the beginning of the 1980s there has been a considerable development in this field. Research in recent times has tended to interpret interactive systems as consisting of a program composed of two independent components, one concerned with the actual application and the other with the HMI. This research has resulted from the appearance of problems related to the absence of an increase in productivity following the arrival of the first interactive systems, and following also the application of ergonomics to try to solve this problem. According to Jurain (1991), large programs pose problems of flexibility, maintenance, and adaptability, and it is therefore important to separate the data handling part of the program from the interface across which communication takes place with the user. The result is that, to some extent, the task of programming can be divided into two parts, one given to those concerned with the data handling and the other to those concerned with interface design. The result has been the appearance of a variety of architectures that have been suggested to separate the application program and the interface program. In Part 3 of this article we will present a collection of ideas about “intelligent” HMI based on the notion of cognitive task and user modelling mentioned earlier, or which have evolved in connection with the most widely used model of interface architecture currently used, the ARCH model. Comprised of five components (figure 2), the Arch model offers an interesting framework for locating "intelligent" HMI concepts. This will be discussed in more detail in section 3.2.

---

[2] One should note that for certain authors such as Coutaz et al. (1992) the word “intelligent” should only be used where the intention is to refer to systems which could pass the Turing test. While we understand this attitude, we will use the term “intelligent” because it has now a well-understood extended use, which amounts to an independent discipline, as evidenced by congresses, meetings, etc., dedicated to its application.

A version of this paper has been published in: *International Journal of Human-Computer Interaction*, 10 (3), pp. 193-231, 1998.

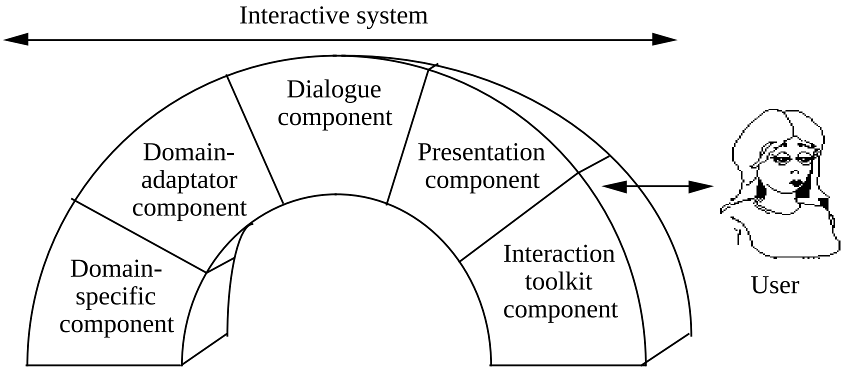

Figure 2. The ARCH Model

It will not be possible in this article to cover all kinds of machine-aided tasks. We will be concerned mainly with operational tasks, that is, those in which the possible activities are well known in advance, and where there are relatively few unforeseen incidents, and with tasks where there is a certain amount of flexibility left to the users to make decisions, whether they make them alone or with decision aids (Rasmussen, 1983 ; Tarby, 1993). We exclude essentially creative tasks and design tasks. Furthermore, cases of programming by example, where there is a major element of machine learning will not be covered. (The interested reader should consult Myers, 1988 and Myers et al., 1993).

2. COGNITIVE MODELS OF HUMAN TASKS AND USERS IN RELATION TO FUNDAMENTAL PROPERTIES AND TOOLS FOR INTELLIGENT HMI

The results of the past 20 years of work in this area has resulted in a number of models, methods and concepts which have been and will continue to be used in the development of "intelligent" interfaces[3]. Two kinds of models are particularly used in the design of intelligent HMI: models of human tasks and models of the user.

It is interesting to note that within the discipline of artificial intelligence one finds more and more often that models of the task itself are being introduced in connection with models of knowledge, for example, approaches to the construction of task-oriented knowledge systems (Chandrasekaran, 1987; Vogel, 1988 ; Wielinga et al., 1992 ; 1993). There have however been few attempts to make a direct connection between modelling tasks in artificial intelligence and modelling tasks in the development of interactive systems, although there are interesting points in common.

---

[3] Note that in future we will for convenience drop the quotes from the word "intelligent", without any intention of adopting a particular philosophical position.

2.1. Analysis and models of human tasks

An increasing importance has been given to the analysis and modelling of human tasks during recent years, both in computer science, and in cognitive science for the purpose of developing interactive programs (see, e.g., Diaper, 1989; Carter, 1991; McGrew, 1991; Lim et al., 1992). In cognitive science, and also in ergonomics as well as work psychology, such methods have existed for many years. Discussions of “task analysis”, and associated methods and references, can be found in many references (Wilson and Corlett, 1990; Amalberti et al., 1991; Sperandio, 1991; Theureau, 1992; Kirwan and Ainsworth, 1992). In addition, there are also certain techniques found in artificial intelligence which can be used as methods for task analysis, such as those described in Dieng (1983) or Cooke (1994). There is a particularly interesting paper by Benysh et al. (1993) which discusses these techniques in relation to the design of HMI.

The reason that these techniques are of great interest for the design of intelligent interfaces is that they have become more and more associated with modelling. These techniques can be automated to a certain extent, and can then be used as tools by those who are developing intelligent interfaces. There are now many approaches that have been developed in engineering and cognitive science for modelling humans performing tasks and which show considerable promise for our domain of interest. In this paper we obviously cannot discuss all such models. They include HTA (Duncan, 1981), DIANE (Barthet, 1995), knowledge blocks (Boy, 1989), JSD* (Lim et al., 1992), TAG and ETAG (Payne and Green, 1989 ; Tauber, 1990), SADT/Petri (Abed and Angue, 1994), UAN (Hartson and Gray, 1992), and so on[4]. In this paper we simply use their ideas in the context of intelligent interfaces. The most widely known of them includes the following.

- The best known model is certainly GOMS (Card et al., 1983). It can be used to model behaviour at different levels of abstraction, from the whole task to physical actions which represent goals, elementary actions themselves, methods to reach a goal, and rules to select methods; it is often used to evaluate design alternatives (Kieras, 1988; John, 1995). GOMS was originally validated for simple office automation tasks, but has recently been applied to more complex tasks (John et al., 1992; Gray et al., 1992; Irving et al., 1994). Even if it is interesting to note that while GOMS has been applied to real-world tasks, the effect has been to reduce their complex tasks into simple, routines - ignoring the richness of the actual work (see for instance Muller et al., 1995). Providing that tasks can be described formally, GOMS can be used to integrate them into an intelligent interface. For instance, using GOMS, possible methods are associated with each task; selection rules express the choice of a method when there is a conflict, i. e. when several methods lead to the same goal. This idea can be used with regard to intelligent interface design, where the interface identifies automatically

---

[4] Readers interested in the full range of models should consult, for example Senach (1990), Balbo (1994), Grislin et al. (1995), Grislin and Kolski (1996).

methods to propose to the user(s) in accordance with their profile, preferences, the gravity of the situation, and so on.

- Because of their formal properties, so-called “linguistic models” (such as ALG and CLG) can also be used in the development of intelligent interfaces. For example, ALG (Reisner, 1981; Reisner, 1984) can be used to represent users’ actions as a grammar, with production rules such as IN ORDER TO do such and such an action PERFORM such and such operations. ALG cannot, however, deal with dynamic tasks and input-output relations, both of which add greatly to the complexity of tasks found in operational contexts. CLG (Moran, 1981) allows one to describe a system at different levels such as task, semantics, syntactic relations, interaction and components. However there are limitations such as a lack of help to structure the model and the lack of any way to represent parallelism or interrupts, both of which are essential in modelling real tasks.

- MAD (Sebillotte and Scapin, 1994), was developed during work at INRIA (France) with the intention of describing human tasks in the light of the need for ergonomics in the design of HMI, and makes use of the paradigm of hierarchical planning which is well known in AI (Sacerdotti, 1977). In MAD a task is represented by a generic object consisting of an initial state, a final state, a goal, pre-conditions and post-conditions. Figure 3 shows MAD applied to air traffic control (El Farouki et al., 1991). On the left is a decomposition of the task “Control the position” in the form of a tree. On the right is the relevant description. As it is at present implemented, MAD can also provide important ideas for the development of intelligent HMIs. (For an example, see 3.3.3).

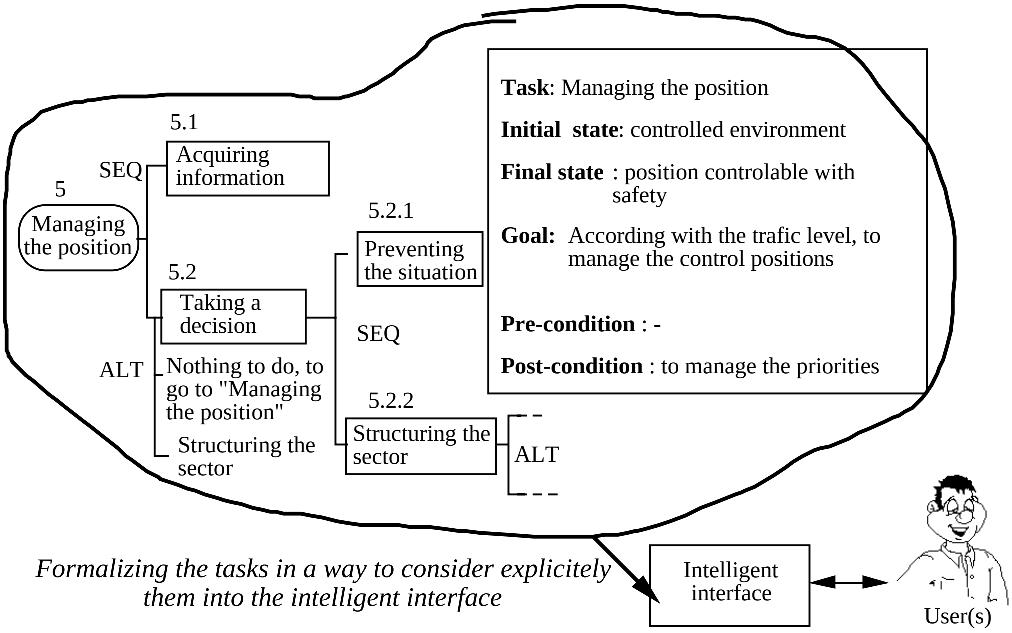

Figure 3. Description of Human tasks and intelligent interface; example with MAD

These approaches of the human tasks modelling have something to offer to the design of intelligent interfaces, as described in Section 3 below. However, they are not sufficient, and must be associated to other kinds of models: the models of the user. Section 2.2 considers this point in more detail.

2.2 Analysis and modelling of the user

Whatever approach is adopted to the design of intelligent interfaces, it is necessary to analyse and model the characteristics of the user, his or her way of working, reasoning, and acting, in order to improve the system and make it more reliable. Thus all users must be characterised by their roles and functions as well as by the kind of artificial aid which is used to assist them. They must all be also characterised by their level of training, their knowledge, and their experience in relation to their ability to understand the complex system in which they are working, both in normal and abnormal operations. Goals and decision criteria must be identified in each situation.

It is obvious that an intelligent interface must be adapted to the way in which operators work and think, and hence one needs to identify the latter. For this purpose there is much literature on which one can draw, which has been developed during the last fifteen years or so.

2.2.1 Cognitive architectures: Human processor model, ACT-R, ICS... Users must frequently make use of their memory in order to carry out their tasks. For this reason many cognitive models have been developed. The usual starting point is to identify three kinds of human memory, sensory memory, short-term memory, and long-term memory (Lindsay and Norman, 1980). Sensory memory stores an image of information about the world as captured by the sensory receptors. The image lasts for some 200 milliseconds in the visual system or somewhat longer in the auditory system. The operator has no control over the loss of information from this image. Short-term memory typically allows one to retain information for several seconds, during which the operator constructs an interpretation of the events using information in the sensory store. The capacity of short-term memory is limited, and although there are reasons to doubt the accuracy of the estimate, it is often related to the “magic number 7 ± 2” as cited by Miller (Miller, 1975). Long-term memory can be thought of, for our purposes, as a complex collection of knowledge, stored as cognitive schemata. Its capacity can be regarded as unlimited. Whatever loss or degradation of memory occurs is not due to limits on the capacity of long-term memory, but rather to the way in which information is encoded and organised.

Based on the experimental study of memory, several models of human memory have been developed, both symbolic and connectionist. The best known is probably that of Card et al. (1983), called “The Model Human Processor”. It is particularly concerned with temporal aspects of human-machine interaction so as to predict the performance of human-machine systems. The authors suggest that someone using a program can be modelled as a system of rules which govern information processing, and which make use of a system of interconnected memories and processors. Each memory is characterised by three parameters: capacity, persistence (length of time for which information is stored), and the kind of information stored. The information that is stored is available for the use of processors outside the memory. Finally, each task can be decomposed into elementary operations. These operations are related to the expected memory requirements of the user in such a way that the need for assistance can be identified. This approach has resulted in several approaches to intelligent interfaces which invoke the notion of “operator assistants” (Cf. 3.3.4).

The notion of “short term memory” has been replaced in recent years by the notion of “working memory”. Baddeley and Hitch (1974) have shown that people can retain up to six items in short term memory at the same time as they are performing operations such as learning, comprehension, or even reasoning. Hence their results are in conflict with the previous model. Consequently, several models have appeared. The best known is the symbolic model ACT* (Adaptive Control of Thought) proposed by Anderson (1983) and subsequently refined (Anderson, 1993). Working memory represents the current state of affairs and the results of information processing, and is related to a long-term declarative memory and a long-term procedural memory.

In addition to models which concentrate on memory there are others, such as the model called ICS (Interacting Cognitive Subsystems) developed by Barnard (1987) (see also Barnard and Teadsale, 1991) and frequently used in discussions of design. This model considers sensory processing subsystems, information processing subsystems, and motor subsystems. In this kind of modelling, a task is described by identifying the subsystems involved in its performance.

Many other models could be cited, including, for example, SOAR (Newell, 1990) or PUM (Young and Whittington, 1990). Our intention here is not to be exhaustive, but to underline the fact that in all cases, the approach to the development of intelligent interface by modelling human performance leads to the need to consider problems. These problems include memory and information processing and their impact on the role of ergonomics in the design of HMI (Kolski, 1997). They are of increasing importance in computer modelling of the user’s cognition.

2.2.2. ***Modelling human activity: The Decision Ladder, Theory of Action, RPD model and Activity Theory.*** To design an intelligent interface, one must also consider how human activity in general can be modelled. The currently most popular approaches are certainly those of Rasmussen, Norman and Klein, and also the so-called Activity Theory.

In 1980 Rasmussen was the first to propose a frame of reference which has come to be called the “decision ladder” which provides a general description of operator activity in a control room. Rasmussen’s “decision ladder” has several sequential levels of information processing (Figure 4a). These include:

(1) The detection of abnormal events which alert the operator following an alarm or the observation of one or more variables with abnormal values;

(2) The observation of the ensemble of variables which allow the operator to identify the state of the system given the previous system state and the goals of the operator;

(3) The identification of a general strategy which will tend to correct the system;

(4) A set of procedures to carry out the actions and the actions themselves.

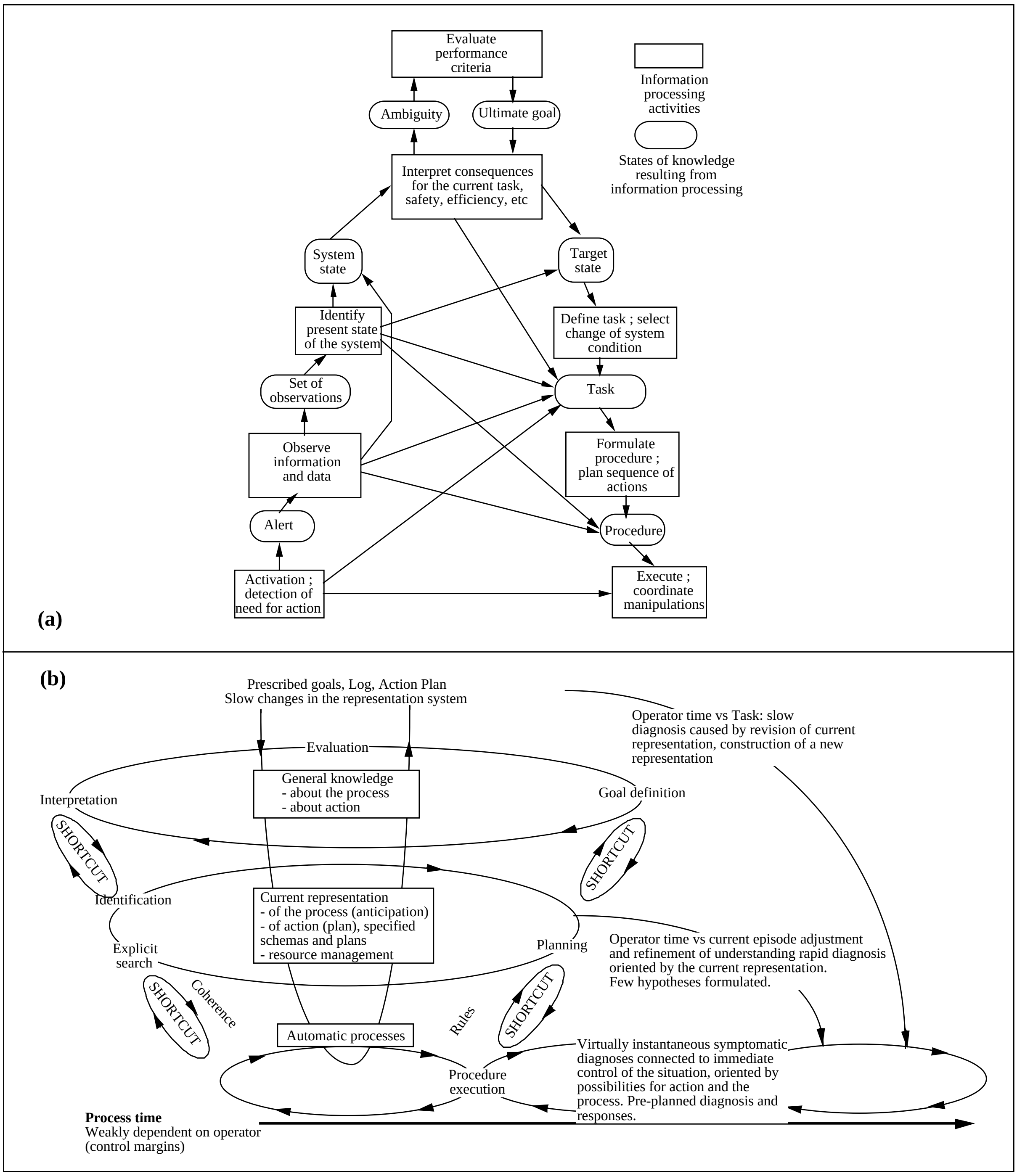

Figure 4. (a) Rasmussen's Decision Ladder (from Rasmussen, 1983, and simplified), (b) extension of the Decision Ladder, considering a temporal dimension (from Hoc and Amalberti, 1995)

The “decision ladder” provides a useful frame of reference for analyse human cognitive behaviour and hence to identify a HMI relevant to each stage.

Rasmussen (1983) identifies three kinds of behaviour:

- The first is skill-based behaviour, the result of long practice, in which the user responds almost automatically with corrective actions in relation to information perceived as signals. These signals provide information about the current state of the environment and allow the experienced user to respond with over-learned automatic actions. The user thus goes directly from an alerted state to the execution of a procedure.
- The second kind of behaviour is rule-based, in which the operator, having identified what state the system is in, chooses an appropriate rule from a set which then identifies the appropriate response. Boy (1986) suggests that the knowledge extracted from experts in order to implement expert systems is essentially of this type.
- The third kind of behaviour is knowledge-based behaviour and is used by the operator when confronted with new or unforeseen situations. In such cases the decisions depend upon really intelligent reasoning, the generation of hypotheses, attempts to verify them and to predict the evolution of the system following whatever attempts are made to manage the situation. The procedures that are used tend to be memorised and used as rule-based behaviour if the situation recurs.

This approach to modelling suggests the idea of cognitive economy. The idea is that past experience produces a cognitive organisation which allows people to use adaptive behaviour requiring a minimum of mental workload in well-known and over practiced situations (Falzon, 1989). So, in any particular situation, and as function of tools and aids which are available, the user will adopt the least demanding behaviour, and thus will use behaviour which is the best adaptation and requiring the lowest possible workload. [5]

Rasmussen’s “decision ladder” is a model characterised by a general cognitive architecture. Such an approach has often been criticised as being too reductionist, too linear and too fixed, in so far as

---

[5] Note that similar ideas frequently appear in research on advanced HMIs which is complementary to the idea of adaptive interfaces discussed in this paper. In particular one can cite the notion of “ecological” (in the sense of Gibson, 1979) displays. From the principles issued from Rasmussen (“decision ladder”, different types of behavior...), Vicente and Rasmussen (92) propose that such interfaces make use of powerful relations between the human and the environment in complex systems. Because people are particularly efficient when acting at a low level in skill-based behavior, these authors suggest that one should present information in a as to avoid the need to work above the level of skill-based behavior. These researchers suggest that interfaces should represent the process at several different levels, from the most detailed to the most global, so that at whatever level is required the operators can find information by “direct perception”. There is a considerable amount of research on advanced direct perception displays (Cf. Vicente, 1995, Moray et al., 1995; Sakuma et al., 1995; Watanabe et al., 1995).

it does not reflect the dynamics of the problems which have to be dealt with. These reasons have led to the creation of derived models. Among those currently concerned with this kind of research are Hoc and Amalberti, who have proposed a new approach to modelling which includes diagnosis and decision making in dynamic situations. They have proposed a model involving the temporal dimension (Hoc and Hamalberti, 1995).

This model proposes three dimensions:

(1) the sequential/parallel dimension;
(2) the autonomy/dependence dimension in diagnosis, in which hypotheses can be selected which are compatible with available actions;
(3) the “delayed/immediate” dimension, which emphasises that diagnosis is not only reactive but also predictive. Hoc and Amalberti suggest that this model can generate a large class of models. It seems very promising, and it would be very valuable to have it validated either in field studies or in a simulator. It seems very likely that research in this area could support work on intelligent interfaces.

With regard to modelling the human activity, another popular approach is the Recognition-Primed-Decision (RPD) model of rapid decision making (Klein, 1989; 1993). The RPD model explains how people can make decisions without having to compare options. Klein considers three cases. In the simplest case, the situation is recognised and the obvious reaction is implemented. In more complex cases, the decision-maker performs some conscious evaluation of the reaction. In the most complex cases, the evaluation reveals flaws requiring modification, and the option is judged inadequate and rejected in favour of the next most typical reaction. Klein explains that "because of the importance of such evaluations, the decision is primed by the way the situation is recognised and not completely determined by that recognition". The RPD model offers also a methodical basis for the design of intelligent user interfaces and decision support aids. For instance, Weiner and Mitchell (1995) consider the RPD model and ideas issued from Rasmussen (1981; 1984), and propose an architecture for understanding fault diagnosis in supervisory control systems. This architecture is called FIXIT (Fault Information Extraction and Investigation Tool); in the FIXIT architecture, case-based reasoning (Kolodner, 1993) is used to encode expert knowledge and experiences, with additional structure for fault management.

Norman’s theory of action (Norman, 1986) introduces the idea of a conceptual model. In fact, Norman calls the conceptual model held by the designer the *Design Model*, and the conceptual model formed by the user the *User's Model*. He explains that the user develops a mental model of the system - the *User's Model*. The User's model is one whose contents form a mental model and whose components are psychological: to each concept, knowledge unit or subject of interest there is a corresponding psychological variable. Norman describes the cognitive stages through which it is necessary to go in order to carry out a task; his model of action has seven stages:

(1) the setting up of a goal;
(2) the formation of an intention;
(3) the adoption of a plan of action by the user;
(4) the execution of the plan of action;
(5) the subsequent perception of the new state of the system as a result of the action;
(6) interpretation of the changes in physical variables expressed in psychological terms;
(7) a comparison of the state of the system with the goal and the intentions previously adopted, which results either in the adoption of the current situation or in further modification.

It is also envisaged that this linear sequence is not always followed: steps may occur in different orders, or be omitted, or repeated. It is clear that in this model the HMI plays a role of fundamental importance. The user is constantly constructing and updating his or her mental model. To be achieved, this modelling action requires the use of the interface, through which the system is represented to the user. Furthermore it is through the interface that the mapping between mental and physical variables takes place.

Activity Theory is a psychological theory with roots going back to the 1920s in the Soviet Union (see Leontiev, 1974). Activity Theory incorporates notions of intentionality, history, mediation, collaboration and development in constructing consciousness (in relation with an activity), and tries to understand the interaction of the individual, other people and artefacts in everyday activity. An international community of researchers is currently applying Activity Theory as a framework to understand the unity of consciousness and activity in human-computer interaction (see the recent volume edited by Nardi, 1995). These researchers have not proposed a concrete architecture or model derived from Activity Theory, and useable directly in intelligent interface design. Nevertheless, a promising research method would consist of studying how the more recent results in Activity Theory and human-computer interaction could influence the architecture and the functions of an intelligent interface. In particular how could the intelligent interface understand (or learn) the *user consciousness* in a way to better assist user activity ?

Every approach to modelling provides an important source of ideas and learning for designers of intelligent HMIs. They provide a way better to understand the needs of users confronted by a variety of different tasks, in different situations, both normal and abnormal, of the system with which they work. For example, in the supervisory control of complex systems one can use Rasmussen’s ladder to indicate different kinds of aids, which might be useful for different kinds of tasks, depending on the level at which the user is operating. (See Figure 5, adapted from Millot, 1988).

A version of this paper has been published in: *International Journal of Human-Computer Interaction*, 10 (3), pp. 193-231, 1998.

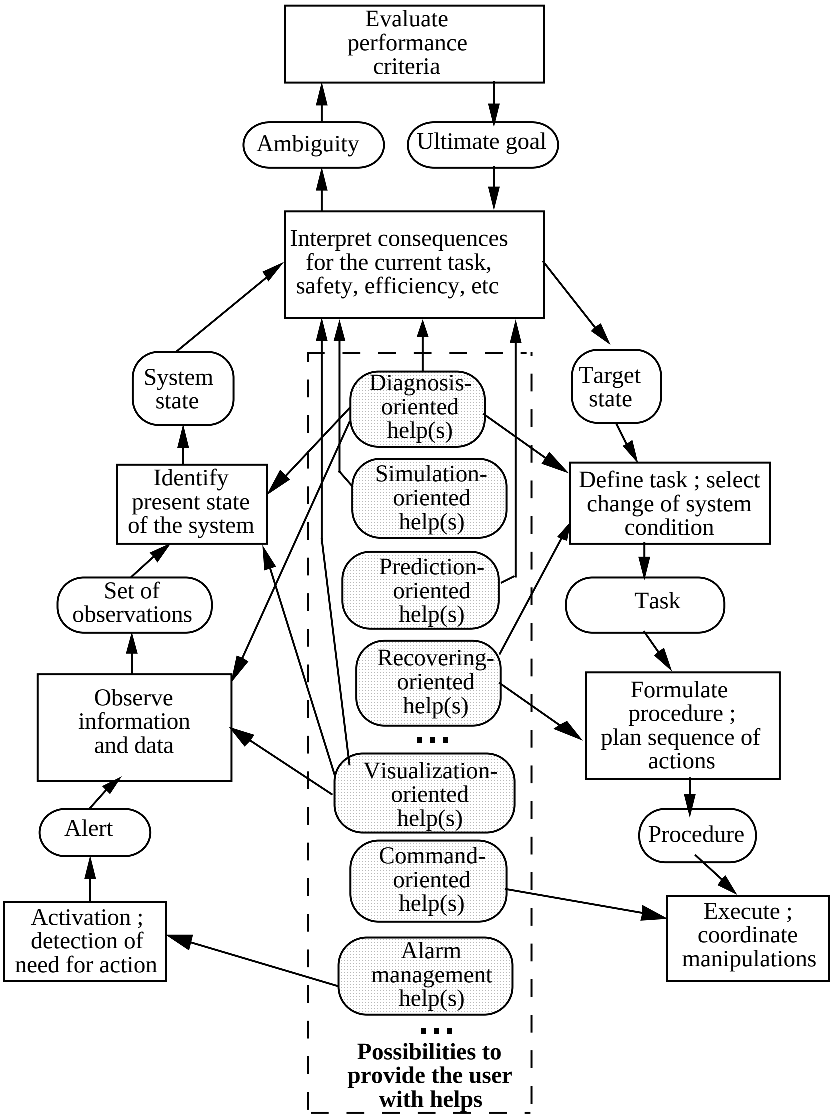

Figure 5. How different kinds of aids can be used for different aspects of tasks

2.2.3. ***Attempts at computer models of cognition.*** Important ideas can be taken from attempts to use computer programs for cognitive modelling. For example, early programs modelled human behaviour in well-defined tasks. They were thought of as normative (Boy, 1983), since the simulated human performance was measured against a well defined norm of possible human-machine system behaviour. However Millot (Millot, 1988) reviewed different kinds of models, and proposed that too often such models were based on an oversimplification both of the strategic decisions and of the overall goals of the operator. In spite of such simplification these early models were very useful both to study and to improve the ergonomics of human-machine systems and to predict critical aspects of their performance.

A second class of models is composed of those which provide psychological explanations (Boy, 1983). Starting from a task analysis they try to simulate the human activity during information acquisition, information processing, decision taking, task execution, and in general the entire gamut of human performance in problem solving. Strategies of behaviour are usually treated as elements stored in a database. A good review can be found in (Bersini, 1989), including a summary of how AI techniques are used to study behaviour in human-machine systems.

Literature provides several examples of applications based on models of cognition. Among them are MESSAGE (Boy and Tessier, 1985) (which models the behaviour of the pilot and copilot of an Airbus 310, see 3.3.4 below); OFMSpert (Rubin et al., 1988), CES (Woods et Roth, 1987) and the model developed from Yoshida et al. (1995) for problem solving in complex systems. One of the most recent attempts to model cognition is COSIMO. It is based on the notions of Reason (1987) and was developed at the Ispra Research Center in Italy (Cacciabue et al., 1990b). The aim of the project is to simulate in a program the cognitive behaviour of operators which is particularly relevant to the genesis of human error and which leads to abnormal incidents and accidents (Masson, 1994). The program architecture contains two levels of reasoning and decision making. There is a “high” level which exploits the ability of humans to recognise (identify) situations in diagnosis, and to use such recognition to plan and choose strategies. In addition there is a lower level, which represents pre-programmed actions, and strategies intended for a well-defined goal or intention. The program then executes and optimises the selected strategy. The underlying concepts include, at the level of perception, “physical salience” and “cognitive salience”, and at the level of problem resolution “similarity matching” and “frequency gambling” (Reason, 1987). COSIMO is written in LISP. It uses a Blackboard Architecture (Nii, 1986a; 1986b) to simulate the set of cognitive activities in an integrated structure in which the simulated operator can pass from one aspect of the task to another in an opportunistic manner during problem solving. This model is still being developed, and represents an important advance. It has already had a considerable influence (see below Section 3.3.2).

In conclusion, it is important for those working on intelligent interfaces to follow closely work on cognitive modelling. As Cacciabue et al. indicate (1990a), such models can provide important directions for further development, particularly for systems which have not as yet been built and which therefore cannot be studied. This work can thus contribute to the design of decision aids. They can also help to understand the nature of cognitive and decision making mechanisms in complex environments.

2.3. Conclusion on the use of cognitive models

Part 2 briefly reviewed the cognitive models of human tasks and users. These models constitute the background of the “intelligent” HMI development, topic of Part 3:

- the “intelligent” interfaces use knowledge of the tasks the user must perform on the process (or application) and thus, their designers need models and analysis methods (like those shown at point 2.1) to represent and manipulate such knowledge.

- the “intelligent” interfaces are designed with the aim of adapting themselves according to their users: the adaptation can include only the user’s preferences, or it can take into account more cognitive characteristics, like the way the user reasons and acts, etc. It is typically on these subjects that the approaches described in point 2.2 can be applied.

3. FROM “FLEXIBLE” TO “INTELLIGENT” INTERFACES.

At present research is beginning to recognise the importance of human factors in the development of advanced and/or intelligent interfaces, particularly in connection with the dialog between the operator and the various cooperative “agents” in the complex system. Several papers have discussed and modelled such a role in the context of the ARCH design model, a generic approach to the description of human-machine interfaces. It is useful to consider some details of the notion of an “intelligent” interface.

3.1. Basic taxonomy of “intelligent” interfaces.

In what follows, we define an “intelligent interface” as any human-machine interface that contains components, which make use of the properties of Artificial Intelligence. Thus we would apply the term to any interface which makes use of, or includes, a knowledge base, a planning mechanism, or heuristics. Equally, we include in our definition interfaces which make use of concepts relevant to Distributed Artificial Intelligence, including functions embodied as agents, including intelligent agents, autonomous agents, intentional agents, etc.

Most interfaces have an important characteristic in common, namely adaptability. They differ, however, in how adaptability is achieved: figure 6 classifies five main types of systems by their degree of intelligence. A flexible interface (which we do not here consider to be inherently intelligent) allows adaptation to the preferences of the user, and according to the system in which it is used, while an error-tolerant interface takes account of the behaviour of the user. An adaptive HMI, in itself, should take into account these two approaches, but generalise them, and adapt itself to the cognitive behaviour of the user. An Operator Assistant, while having in principle the same abilities as an intelligent interface, has further levels of autonomy, and behaves almost like another human assistant. An intelligent agent has in principle all the above characteristics, but in our opinion represents the arrival of a truly “Intelligent” HMI because of its ability to model cooperative human-machine systems, or even socio-technical systems (Le Strugeon et al., 1995).

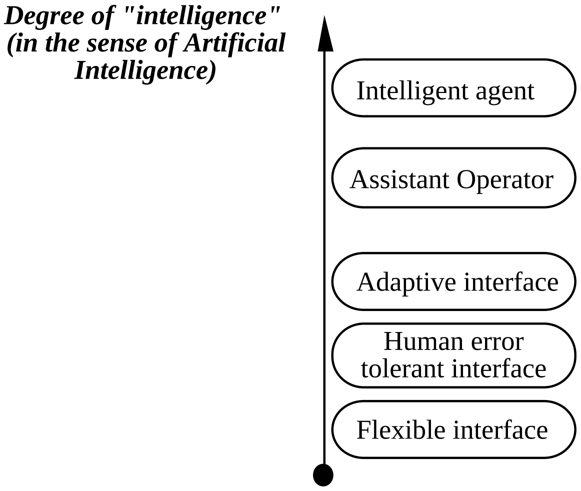

Figure 6. Types of intelligent interfaces

According to the “law of requisite variety” (Ashby, 1956; Conant and Ashby, 1970), more complex systems require more complex controllers. As a result, the interface implementation complexity increases with its degree of intelligence. Most intelligent interfaces are based on artificial intelligence techniques to model users from the point of view of their cognitive activities. Current research proposes to develop interface architectures based on knowledge of users’ cognition. Although such research has hardly begun, promising concepts and models have already appeared. It is therefore worth pointing out some of these possibilities for the future.

We first deal in section 3.2 with the models of architecture that underlie the description of the intelligent interfaces to be found below in section 3.3

3.2. Brief review of models of interface architecture.

Several models of HMI architectures can be found in current literature. [6]

A very general model is LANGAGE, proposed by Foley and Van Dam (1982). LANGAGE starts from the assumption that both the application and the user interface can be decomposed in the same way as a language. The decomposition provides four levels of analysis (figure 7a):

(1) The conceptual level, where the objects of the application are listed, along with their attributes, their relations, and the actions that can be performed on the objects. This level corresponds to the application:

(2) A semantic level, where the functionality of the system is described, including the set of actions possible to perform on each object, without bothering about the sequential connections between different actions:

---

[6] The interested reader should consult Coutaz (1990), Duval (1994) and Edmonds (1992) who provide examples of intelligent or non-intelligent systems based on such models.

(3) A syntactic level, at which sequences of user actions are defined in terms of input-output relations to perform the actions described at the prior level; and

(4) A lexical level, where one finds primitives associated with the control of input events and with the display of outputs.

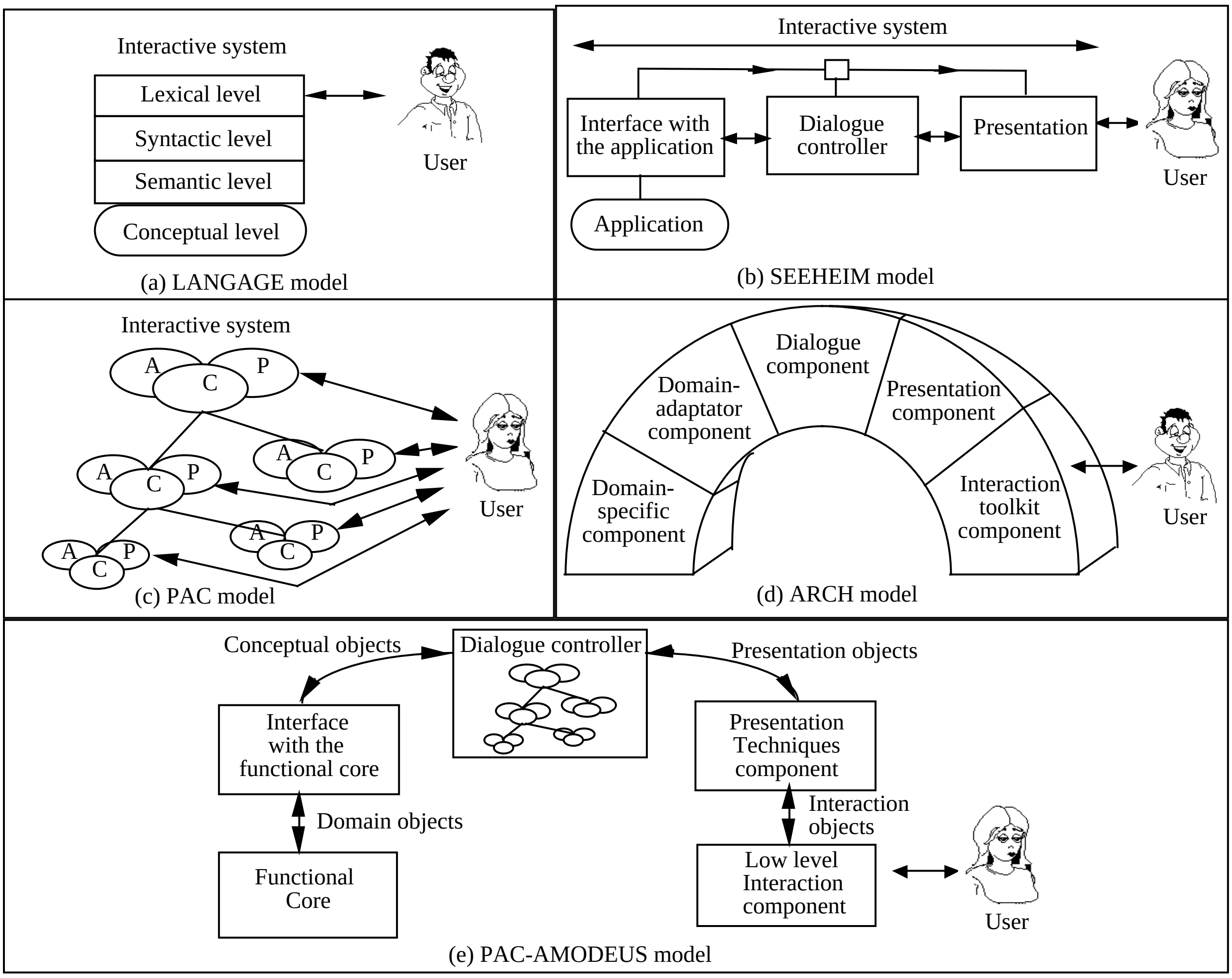

Figure 7. (a) LANGAGE, (b) SEEHEIM, (c) PAC, (d) ARCH, (e) PAC-AMODEUS models

The basic properties of the LANGAGE model have been used by many researchers and have influenced the development of several generic models. The best known is probably SEEHEIM. In the 1980s a SIGGRAPH working group defined a human-computer model called SEEHEIM (Pfaff, 1985), which takes its name from the place (SEEHEIM in West Germany) where it was defined during a seminar on the design of user interfaces. The model has three logical components (see Figure 7b):

(1) presentation which is the part visible to the user and which controls input/output, and which corresponds more or less to the lexical level in LANGAGE;

(2) the “dialog controller” which is responsible for the structure of the dialog between the user and the interactive system, and which according to Coutaz (Coutaz, 1990) corresponds to the syntactic level of LANGAGE, since syntactically correct phrases

correspond to requests or data which the user wishes to exchange with the application, and in reverse the receives abstract commands for output which are sent to the specialised presentation elements;

(3) the interface with the application which has the task of passing data between the interface and the application, within the limits of the semantics of the data and dialog. This level corresponds to the semantic level of LANGAGE. The application itself corresponds to the conceptual level of LANGAGE.

According to Coutaz (1990), while SEEHEIM has limitations due to its generality, it offers several advantages. These include:

(1) a frame of reference for defining the structural bases of the system to be built;
(2) an easy way to perform iterative modification during development since each part can be modified without any effect on the others,
(3) it is not tied to any particular software for implementation, and
(4) it can serve as a starting point for the development of automatic generation of interface characteristics for each of the three components. The SEEHEIM model provides a reference point for software architecture for interactive applications. Several later versions exist, or models derived from it, such as modified SEEHEIM (Dance et al., 1987) which extended the semantic level; extended SEEHEIM (Karsenty and Weikart, 1991), which aimed to include the complete application by means of extra components.

The most commonly cited version today is the ARCH model (Bass et al., 1991; The UIMS Developers Workshop, 1992) (see figure 7d). ARCH represents an interface application as being composed of five components:

- a toolkit that includes interactive objects. This component would correspond to the physical level in a decomposition of the SEEHEIM presentation element;
- a presentation component, that corresponds to the logical level of the presentation element’s decomposition. It includes logical objects (tool-independent) and links the dialog unit to the presentation toolkit;
- a dialog controller, which is the keystone of the arch and corresponds to the dialog controller of the SEEHEIM model;
- a domain-adaptor, the intermediary between the dialog component and the application. It adapts the tasks to the domain and adds the “meaning” (a semantic level) to the data that comes from the application;
- a domain-specific component, which has specific knowledge of the domain, ie. an interface-independent representation of the application.

The PAC (Presentation - Abstraction - Control) model proposed by Coutaz (1987; 1990) makes the assumption that an application and its associated HMI can be decomposed into a hierarchy of interactive agents. The semantic, syntactic and lexical levels are thus distributed throughout the basic entities (called interactive agents). Each entity is made up of three elements (see Figure 7c).

- a Presentation component which defines the exact syntax of the application, that is the visual appearance of the objects, their input/output behavior, and their relation to the user. The figure shows that depending on the level of abstraction being considered, the user interacts with a particular agent. The presentation of the application then corresponds to the union of the agents of which it is composed.
- the Abstraction component which defines the semantics, that is the functionality of the application.
- the Control component which handles the coherence of the relation between presentation and abstraction, acts as arbitrator and decision maker, and looks after any translation needed between abstract and concrete forms of a PAC object.

PAC modelling offers a recursive way to describe an interactive system, because PAC can be applied at each abstraction level. For instance, if at a level i, a PAC agent represents a car, at a level i-1, other PAC agents could represent respectively the tyres, the car body, and so on; at the level i-2, other PAC agents could represent sub-elements of the car body, and so on. In that condition, PAC model corresponds to a distributed version of the SEEHEIM model (and therefore to a distributed version of the ARCH model), with the addition of the notion of an agent. It can benefit from the properties of agents - modularity, the ability to look at the situation at different levels of abstraction, and the possibility of parallelism in regard to input/output relations. [7]

Several researchers are currently trying to exploit the advantages of such a distributed architecture while using ARCH to guide implementation. For example, Nigay and Coutaz (1991) propose a hybrid model called PAC-AMODEUS, combining the components of ARCH with the distributed agents of PAC (figure 7e). In particular the controller of the SEEHEIM model can be made up of PAC agents. These kinds of models of HMI architectures represent a very important advance in the development of interactive systems. It is also important to note that currently many researchers are trying to adapt these models to develop architectures which permit multi-modal interaction (Ferrarri, 1994; Nigay and Coutaz, 1995) or Computer Supported Cooperative Work (CSCW) (Hill, 1992; Dewan, 1992; Croisy, 1994).

Each of these fundamental models stresses one specific aspect of the interactive system:

- the abstraction level in LANGAGE;
- the roles of the components in SEEHEIM and ARCH;
- the hierarchical decomposition in PAC.

However, as we have seen in this section, some similarities can be found between them. The SEEHEIM model can be expressed in terms of LANGAGE levels, just as for the ARCH model

---

[7] In order to keep this article short and clear we can only consider generic models. Thus we do not deal with MVC (Model, View Controller) which is the logical model underlying SmallTalk and which like PAC is agent oriented. The interested reader should consult (Goldberg, 1984).

A version of this paper has been published in: *International Journal of Human-Computer Interaction*, 10 (3), pp. 193-231, 1998.

which is an extension of SEEHEIM, and in grouping together each kind of PAC elements, the three main elements of SEEHEIM can be found again. Finally, PAC-AMODEUS is an illustration of how these different models can be combined to produce hybrid models, which can answer specific design needs.

To illustrate the kind of intelligent interfaces proposed in section 3.2.2 we can use ARCH to organise the presentation of the different approaches. The reasons we use ARCH as the key to our presentation are as follows:

1. ARCH is a model now well known to most computer scientists;
2. ARCH is satisfactory as a frame of reference for all the ideas we wish to present;
3. it will not be difficult for a reader to translate what we say in this framework into other frameworks if required to do so.

In what follows we will try as far as possible to hide features which are related to the particular method of implementation (neural network, natural language, inference engine, hypermedia, etc.) and present the general features.

3.3. Review of the concepts of "intelligent" interfaces in relation to the ARCH model.

3.3.1. Flexible interfaces: "advanced" "non-intelligent" interfaces. The concept of a flexible interface, also called "configurable", "personalisable", or "adaptable", is one where a single dialog support system can be used to communicate with several different systems. In the most favourable case the user can configure the interface and its support according to subjective personal criteria, especially habits or preferences for the operational modes and the form of the human-machine dialogue. In the most extreme cases, which do not seem to have yet turned up in the literature, one could even think of choosing the type of interaction, for example direct manipulation, interaction using speech, natural language, menu, a method suited to the database, etc (Waern, 1989). A flexible interface uses only a very low level of intelligence, and generally consists of a database used by a presentation controller (Williges et al., 1987), which later is guided by the user according to the approach to a particular problem. [8]

Generally speaking the knowledge needed for a flexible interface can be grouped into three knowledge bases (see figure 8). The first knowledge base holds the protocols for communication protocols with different applications that make up the work environment of the user. The second

---

[8] The idea of the flexible interface is currently evolving by taking into account ever more frequently ideas related to human tasks. This is true, for example, in something like MAcWeb which is being developed for an interactive technical documents data base using hypertext. In this system access to information in the documents is possible according to their use in connection with the current task, and not because of relevance to a particular document. This is made possible by a network of relations in a hypertext-based data base (Nanard and Nanard, 1995).

holds modes of presentation (colours, sizes of windows, etc.) and the different styles of interaction which users can choose for the interface. The third is specific to each user of the overall system, and contains individual preferences for modes of presentation and/or styles of interaction.

So, when an application is selected, the user implicitly chooses a communication protocol, which is suitable thanks to the "chooser", for the application. Users then perform a dialogue with the application using an interface which has already been configured but which is sufficiently flexible to be changed simply and frequently as required by the users.

As an example, consider the work of Muller (1988) who developed a flexible interface allowing users access to several AI tools and several databases. The AI tools and the databases were distributed throughout a network of three computers, which were mutually incompatible from the point of view of their operating systems. Muller's problem was to find a way to get at the databases and the AI tools in a way that was transparent to users. Otherwise, it would mean that all users would need to know at least some operating system language commands to get access to each computer. Muller's solution was a flexible interface controlled by an extra computer. If the user wanted access to Computer #1, then the extra computer emulated Computer #1 using its appropriate databases, and similarly for the other two computers. Thus, no matter where users' requests were sent, users could always interact in the same way. The users' tasks were much simplified because they could always use the same instructions and the same HMI dialogue no matter with which computer in the network they interacted. They could modify certain characteristics of the dialog, which were then memorised by the extra computer.

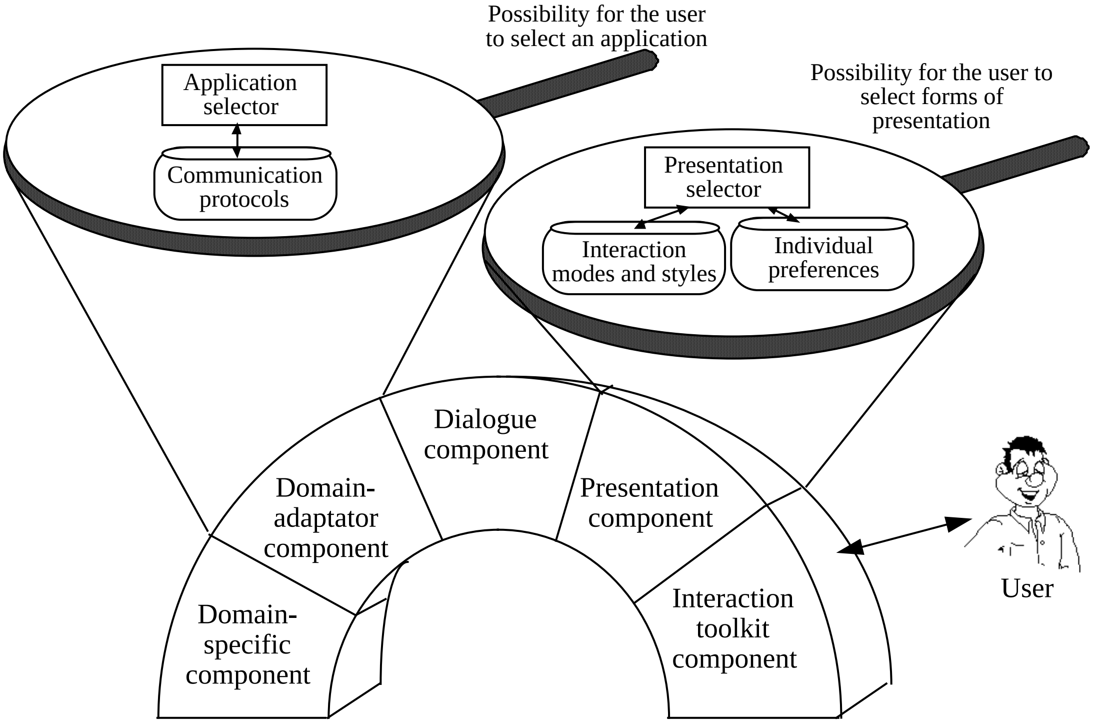

Figure 8. The concepts of a flexible interface /ARCH model

Flexible interfaces offer several advantages. First, the users' actions are filtered and translated into the language of the relevant application. Second, information coming from the application are, in the same way, filtered and translated according to a predetermined style. Third, It becomes much easier to use a mixture of heterogeneous applications. However, it is usually very costly to develop such interfaces, as the cost depends heavily on the number of applications and their degree of heterogeneity.

3.3.2 Human error tolerant interfaces. The development of interfaces tolerant of human errors is in practice based on studies of the kind of errors which humans make in simulated conditions. In these studies errors are identified by recording actions which result in the behaviour of the human-machine system failing to meet well-defined criteria of productivity or safety. The notion is to use such studies to develop ways which, in the real world, will replace, improve, or cancel inappropriate human actions (Rouse and Morris, 1985; Hollnagel, 1989; Beka Be Nguema et al., 1992; Masson, 1994). It should be noted that from the theoretical viewpoint these studies are based on task analysis, and modelling human tasks about how people reason, leading to human error taxonomies (Leplat, 1985; Rasmussen, 1986; Reason, 1990).

The structure of such interfaces usually consists of several major components (see Figure 9). There is a decoding module, which translates users' actions into data that the system can use. A second module identifies the action of the users, using a model of the context of the application to identify prescribed tasks. It can then correct the action if necessary according to criteria such as productivity or safety. A third module explains to the users the problems diagnosed by the system and the advantages to be gained from its proposed intervention.

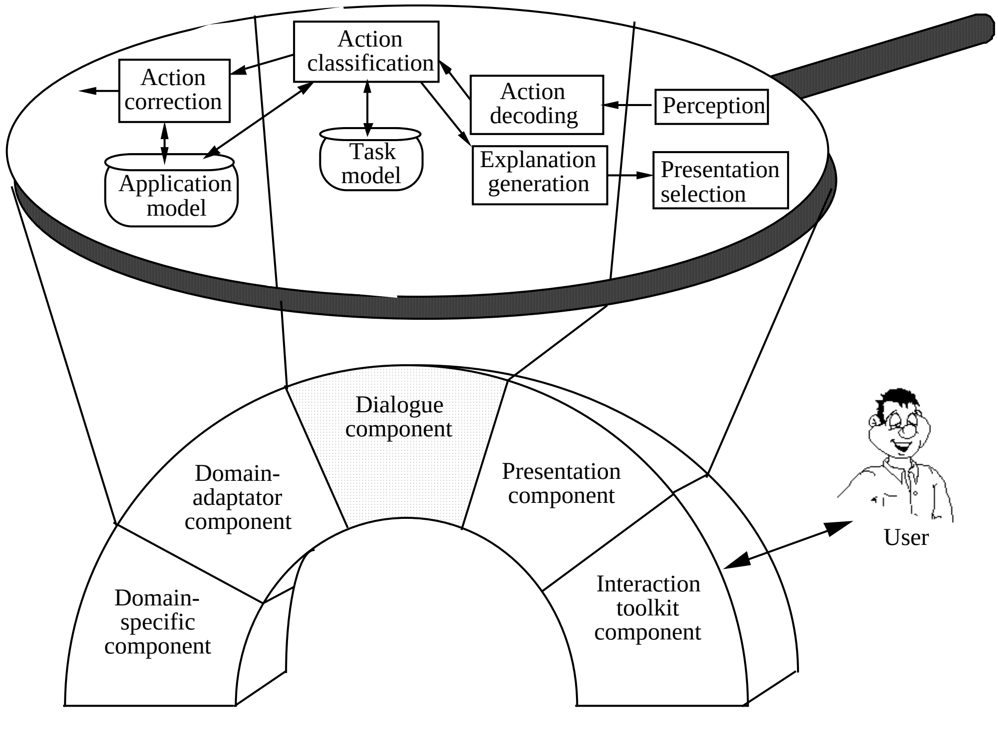

Figure 9. An error tolerant interface represented in the ARCH model

An example of this kind of system can be found in the RESQ system (Hollnagel, 1989; 1994). RESQ can detect human errors on line, in real time, in a well-defined domain. It combines identification and evaluation of plans of action as well as error elimination. RESQ is part of the GRADIENT system (Projet Esprit #857). A goal of that project was to implement "intelligent" assistance for industrial process control, particularly in power generation. RESQ uses plan recognition to help the control room operator. RESQ supervises operator actions and looks for cases where the actions seem incoherent or incorrect in the light of the operators' current goals. (An example is when an operator fails to complete series of actions begun in the past, or when actions are greatly delayed, or several actions performed at the same time.) Hollnagel (1989) states that the system seems promising, and the results show it could be applied to other domains. However, much work remains to be done before it can be implemented widely in industry, and in particular one should be able to profit from recent advances in artificial intelligence, particularly in plan generation. Also, there would be a great advantage if automatic plan learning could be implemented.

The usefulness of such a principle is undeniable, but it cannot always be applied. Such a system required an extremely reliable model of the application, both of the possible situations and of the possible actions. The latter will not be exhaustive, because it will probably be based on well-defined fault conditions. But in reality, totally new situations often arise. In the latter case, neither the error tolerant interface nor the operator will have the necessary knowledge to cope with the problem based on past experience, and this type of system will then not be effective.

3.3.3 Adaptive interfaces. The idea of an adaptive interface dates from the end of the 1970s. The way it extends the ideas of the previous two classes of interface is to add the notion that the dialogue between users and the system should adapt itself to the tasks which users have to accomplish. (See section 3.1 above.) Available literature contains many examples of architectures and methodologies for the development of adaptive interfaces (Edmonds, 1981; Rouse, 1988; Hefley, 1990; Brajnik et al., 1990; Kolski et al., 1992; Schneider-Hufschmidt et al., 1993). For example, Hefley (1990) integrated several approaches to develop a theoretical architecture using a group of agents whose aim is to support interaction between a given application and users (See figure 10).

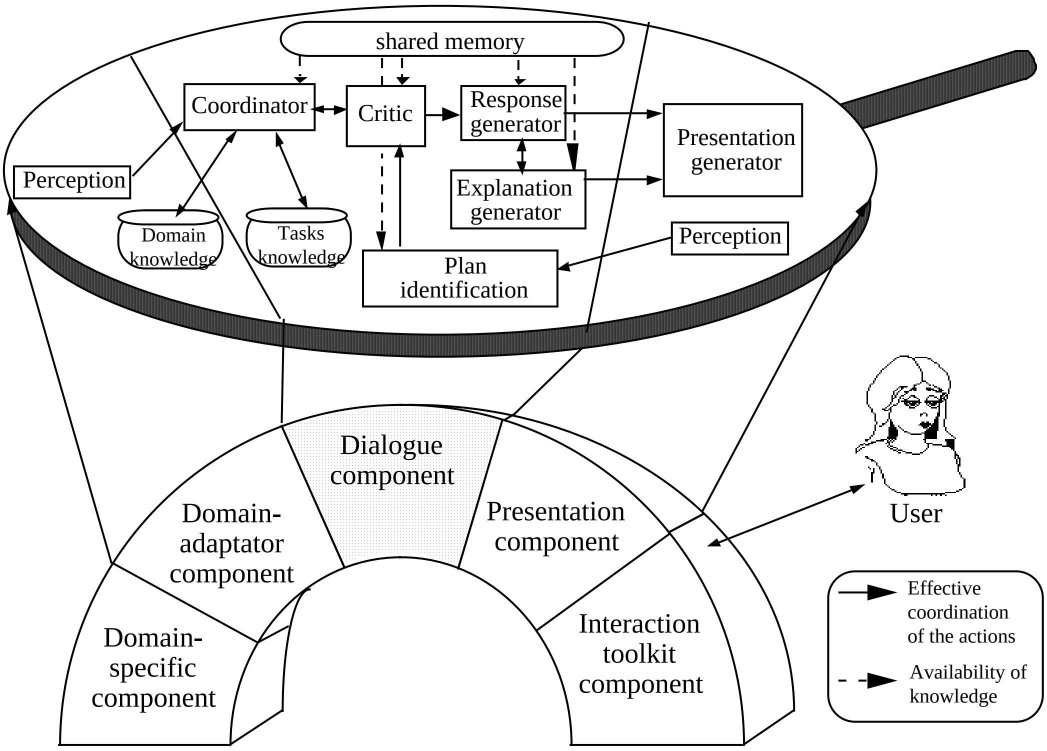

Figure 10. An adaptive interface in the context of ARCH model

The architecture proposed by Hefley contains the following elements.

The plan identifier uses a database of the most common users' tasks and the observed actions of the users to infer the users' goal. The critic has the task to identify users' actions that can be criticised in the light of the constraints to be observed (similarly to the notion of error tolerant interfaces). The coordinator of domain knowledge can assign certain tasks to the application program (such as calculation, table look-up, etc.) if the situation indicates that there may be a problem such as operator mental overload, the need for simultaneous tasks, etc. It uses allocation heuristics and specific domain and task knowledge. The response generator replies to questions put to it by users (including those that arise in unknown situations). The explanation generator has the task of explaining to users the answers it has generated. When asked by users, this agent gives the reasons for its choice of solution (including, for example, why it decided to take over a task from the user). The presentation generator handles the problem of making certain that the way information is presented to users is compatible with their current state of thought and knowledge.

The PAC model can be used to detail these different interactive agents. Each of the elements above can correspond to a PAC agent of the highest level in the architecture, with a specific role in the organisation of the interface. For example, the agent plan indicator determines what goal the user is pursuing based upon the user's actions. At lower levels, each of them can be seen as composed of smaller and more specialised agents. For example, the plan indicator can include specific agents for actions, representation of actions, inferences, etc.

Adaptive HMIs are the subject of much current research, but are turning out to be very difficult to implement especially in complex situations. Several examples can be found in the literature such as Bisson et al. (1992) and Furtado et al. (1995) which are based on MAD as a task model. Other examples can be found in intelligent tutoring systems field.

One example which has been implemented, called "Imagery Decision Module" (MDI in French), can be found in Tendjaoui et al. (1991), and Kolski et al. (1993), in the form of an intelligent system to manage dynamic displays (Figure 11). MDI is part of a supervisory control program that filters information and presents the user with information pertinent with the current state of operation, placing it in a particular part of the screen. (The system is a complex one and includes information about prediction, diagnosis, detection of abnormal events, etc.) MDI is original in that it has three kinds of production rules underlying the interface. The first is WHAT to present to the user. The second is WHEN to present it. The third is HOW to present it. These three modules are connected to others which relate them to aspects of the situation, such as how serious the situation is, what kind of user is currently involved, what kinds of displays are available, and so on. Thus MDI takes decisions about WHAT, WHEN, and HOW in the light of how the situation unfolds, and maintains a permanent image in a specific area of the screen, to provide the user with the best possible and most up to date information.

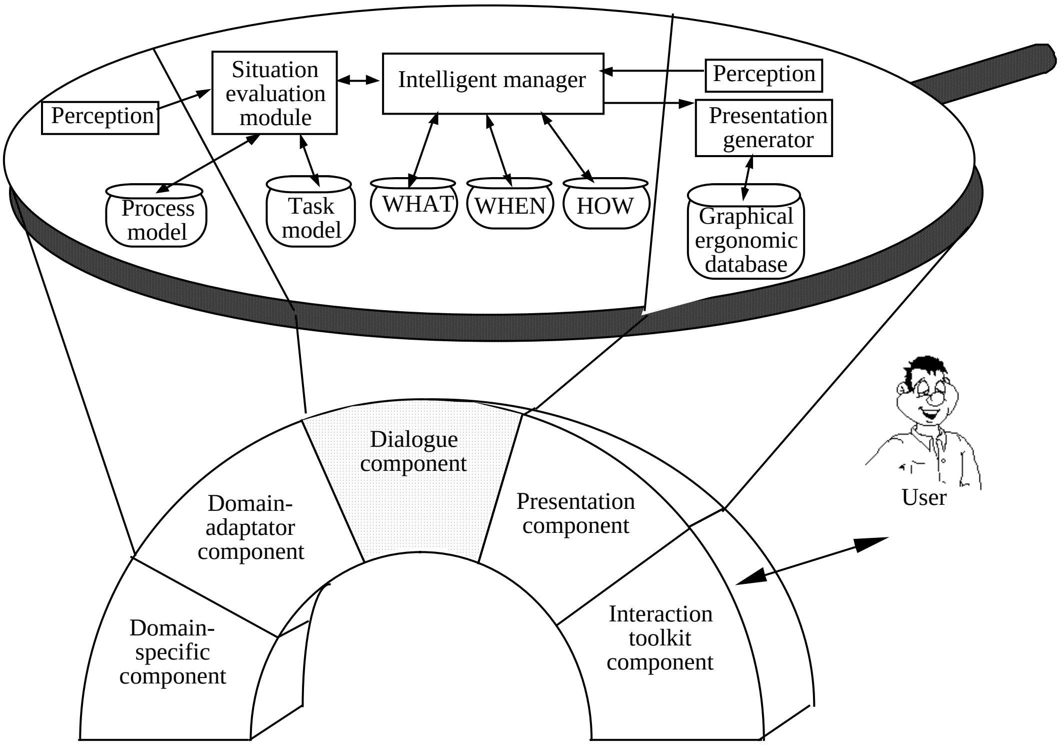

Figure 11. The M.D.I. (example of adaptive interface) in the context of ARCH model

3.3.4. Operator Assistants. In the ”Operator Assistant” approach the interface behaves as if it were a user’s assistant, and helps the user to complete tasks. This approach can be thought of as a generalisation of an adaptive interface. Such a system has both knowledge about the tasks to be performed and the relevant data, and also continuously generates plans to accomplish tasks. The plans are able to do the following: (1) they can be used by users if necessary; (2) they can be immediately used by the Operator Assistant in the program if the user asks for them to be used. The concept of an “assistant” is not as recent as is sometimes thought, since the concept can be found in a paper by Bullinger and Fachnrich in 1984.

Boy is one of the leaders in this area as a result of his work for NASA on “Intelligent Operator Assistant” projects (Boy, 1985; 1988; 1991). He gives the following example:
“In an aircraft cockpit, a human co-pilot shares the work with the captain, but does not have the final responsibility: the captain is the captain! The captain can consult his co-pilot at any time during the flight, but the former has the ultimate responsibility. If the captain delegates part of his responsibilities to the co-pilot, then that responsibility becomes a task for the co-pilot to perform. Furthermore, the captain can interrupt a co-pilot’s task at any time if he thinks it necessary. However, a co-pilot can take personal initiatives, for example to test parameters, keep himself up to date with the development of a situation, predict which tasks can be foreseen, etc. A co-pilot can make use of the instructions in an operating procedures manual to the request of the pilot. He must be able to explain, at an appropriate level of detail, the results of any such use.”

As an example, it is interesting to look at some of the early results of Boy in connection with an interface developed in the context of an operator model called MESSAGE (Boy and Tessier, 1985; Boy, 1986), during the evaluation of a cockpit design. The system being controlled was a simulator of a system for supplying the needs of satellites in orbit using the shuttle. The operator assistant, developed from concepts such as those found in Newell and Simon (1972) and in Card et al. (1983; Cf. 2.2.1) was embedded in a system for fault diagnosis called HORSES. The structure of this assistant can be thought of as a being made up of several modules (see Figure 12) (Boy, 1988). The most important elements were the following. First, a pattern recognition module which took into account the desired situation, the actual situation, situation awareness, and analytic knowledge in a “long term memory”, and which allowed the assistant to choose a subset of rules to use for inference. Second, there was a module for solving problems, which used the rules selected, by the first module, with the aim of helping the user, in response to a request made by the latter.

The knowledge structure allowed the operator assistant as many as six levels of autonomy. In the case of MESSAGE these were: [9]

- Level 0. The knowledge base was not used because the assistant was inactive.
- Level 1. The knowledge base was connected to the application, and the operator assistant could deduce different possible situations. Each situation was regarded as a context in which different actions would be appropriate.

---

[9] Other famous works are related to levels of autonomy: see Sheridan and Verplanck (1978) or Moray (1986).

- Level 2. The operator assistant could deduce one or more contexts that the human operator might choose.
- Level 3. The operator assistant automatically chose a context appropriate to the current situation. If the human user approved that context, the assistant would undertake to execute the appropriate action and would send a message "executed" to the human user. The user could however decide to perform the task using the corresponding actions.
- Level 4. There was completely automatic analytic diagnosis. The operator assistant decided to analyse the situation, identify the context, and inform the user of the result. During this phase, if there were no questions that required the agreement of the user, the latter did not see the process taking place. At the end of the analysis the assistant proposed actions to the user for confirmation.
- Level 5. Everything is automatic. The operator assistant offered advice about the situation as well as about diagnosis. The assistant could also explain what it was doing to the user.

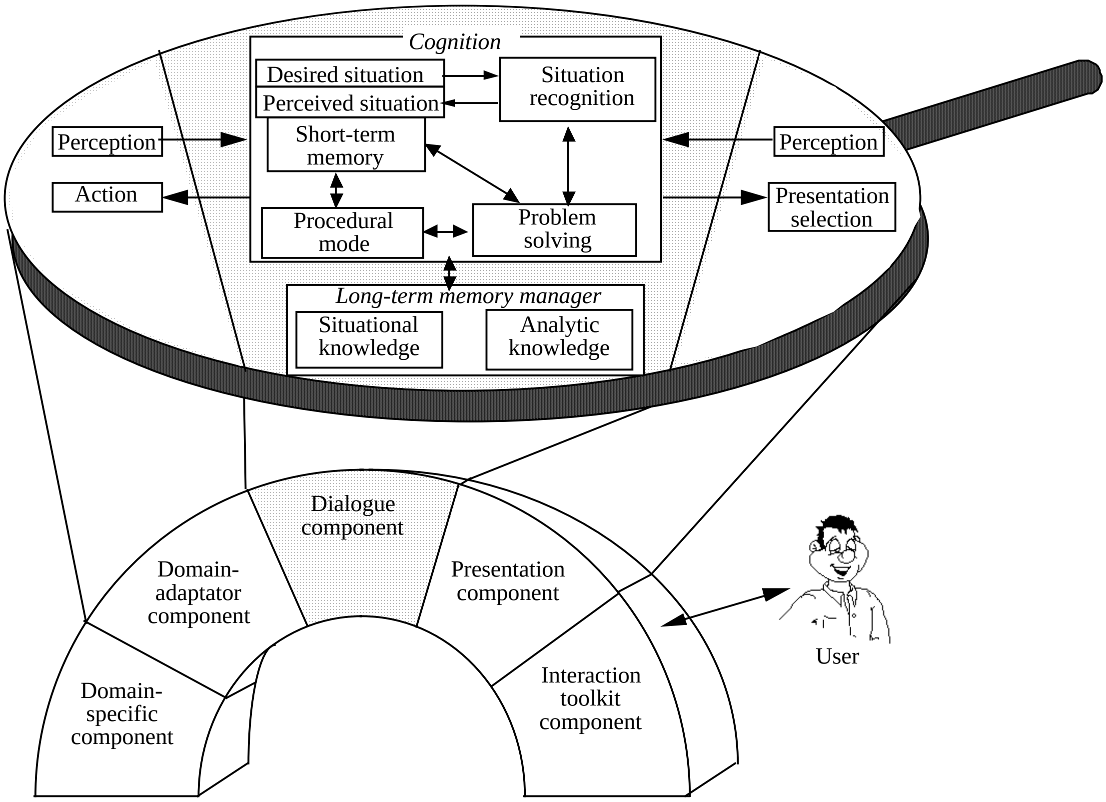

Figure 12. An Operator Assistant in the context of the ARCH model (inspired from Boy, 1988)

We see, thus, that the structure of the knowledge base allows an operator assistant to work at several potential levels of autonomy. Current work is concerned with the development of operator assistants in aviation and in space (Boy, 1990; Shalin et al., 1990; Amalberti and Deblon, 1992; Gerlach et al., 1995). These projects prove the feasibility of the approach, but, of course, much pluridisciplinary research is still necessary to study and optimise the possible cooperation modes between user(s) and operator assistant(s) (Millot and Hoc, 1997). In particular, recent work has added the idea of Machine Learning in such assistants, with the goal of extending, revising, optimising and adapting the knowledge base(s) of an assistant system, and then to improve their utility and maintainability. For instance, the COSIMA system dedicated to floor planning (a sub-task of VLSI design) is based on Machine Learning methods; COSIMA is able to learn to avoid some errors that occur in the problem-solving process performed by the user, by observing the user (Herrmann, 1996).

3.3.5. The HMI viewed as an intelligent agent. There is clearly a close link between the ideas of an intelligent agent and an operator assistant. The concept comes from work on distributed artificial intelligence, and the attempts to use a formal logic to model the behaviour of a rational agent. Currently the model is thought to involve the need to model intentional states, belief, desire, intentions, and undertakings (promises) (Chaib-Draa et al., 1992; Mandiau and Le Strugeon, 1995).

The notion of an HMI using the concepts of an intelligent agent arises from the possibility of decomposing the human-machine system into a set of agents (see Figure 13). These agents would work in parallel or would cooperate, with the goal of solving their relevant problems in the light of the task to be performed. The results of their activities would be transmitted to the users, but at the same time they would perform a large number of other operations, for example to control the system itself. Mandiau et al. (1991a) have predicted the development of such systems, foreseeing (1) complex systems developed as reactive agents, (2) user assistance embodied as cognitive agents, (3) groups of users as external agents, (4) other HMIs viewed as intelligent agents. Thus the HMI becomes embedded in a universe inhabited by multiple agents (Gasser and Huns, 1989; Gasser, 1991).

In such a situation the HMI would not have to, in principle, merely understand and reason, but to take into account reasoning in relation to desires and promises (undertakings) to perform an action or fulfil a need, as well as to understand different goals. Such a module would have to be able to undertake to reach one or more goals that had been selected. If the goal is attainable, then a plan has to be generated; if not, then the goal should be selected by another agent (perhaps one in the domain of competence of a different interface). The plan would have to contain intentions to undertake tasks, and beliefs about the surrounding environment. The module would also include a number of abilities such as communication or probable responses of other agents. After the development of the plan, the agent would execute the plan in order to reach the goal for which it had been invoked. The goal would be achieved by a module that combined task execution and communication, depending on whether one or more acts of communication and cooperation with one or more other agents were needed. Such a module should be able to decide the appropriate agents to invoke to interact with itself to achieve the goal.

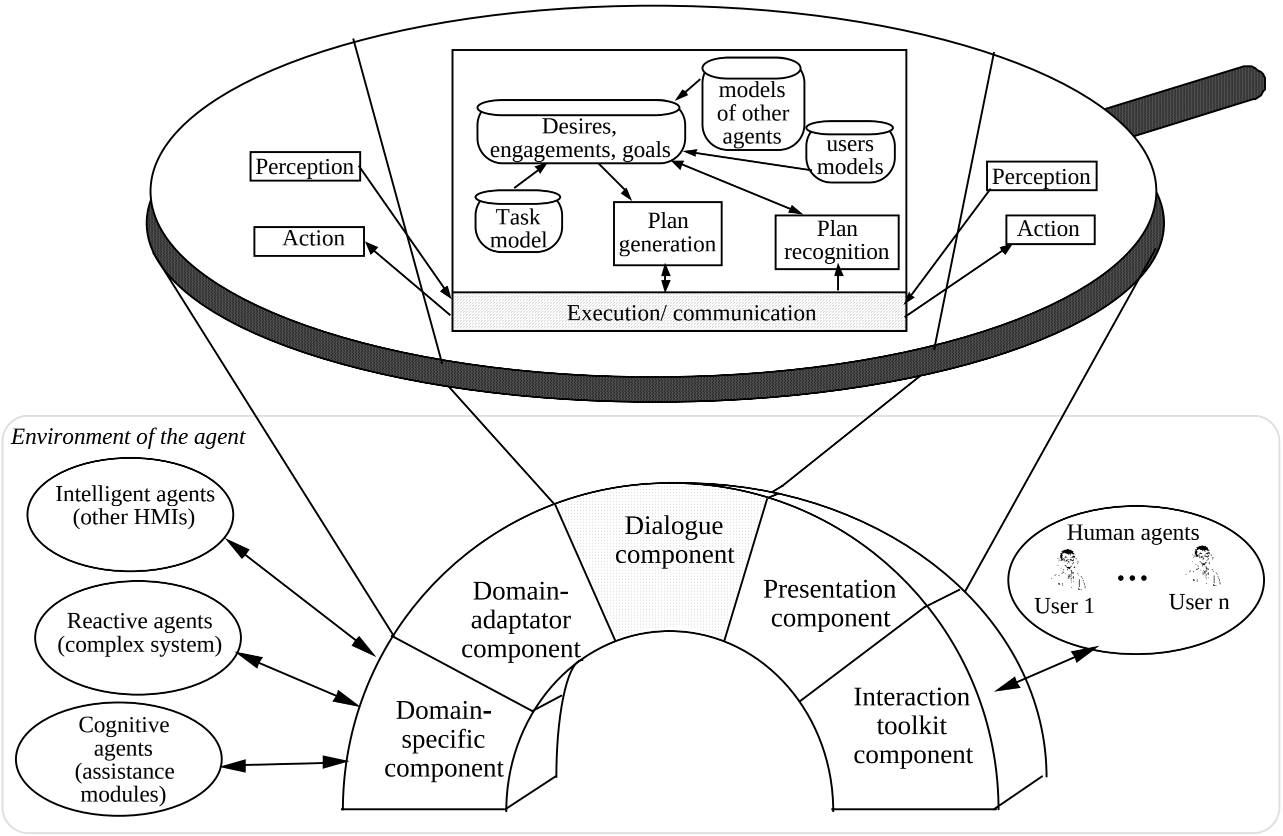

Figure 13. An intelligent agent in relation to ARCH model

It is clear that such an idea indicates the need for a great deal of research and thought. One should consider at least the following three points. First, the need for the user to be able to work in a manner which makes the operations of the assistant “transparent”. Second, how can human-machine systems be designed so that they can be decomposed into agents. Third, how can such systems be evaluated (Mandiau et al., 1991b; Banks et al., 1997)? Finally, there is a major problem of formalising different kinds of knowledge, especially when the number of agents becomes large.

Currently this approach is speculative. Some of the ideas have been implemented in a system using cooperative agents in complex dynamic systems (Avouris and Liedekerke, 1993), developed using an architecture of multi-agents called ARCHON (Wittig, 1992). Fournier (1994) is currently working in the same tradition with the aim of providing computer programs that are more “human” and more “gentle”. Research shows a trend to building interface agents seen as “personal assistants”. Such agents assist the user in his or her tasks. Most of them are based on a learning approach: they observe the human-machine interaction to learn how they can, then, assist the user in regular activities (see, for example, Maes and Kozierok, 1993; Maes, 1994). A number of the Communications of ACM in 1994 were concerned with intelligent agents in interactive systems. It is clear that these approaches are becoming more and more popular for the development of cooperative systems (see for example Nakauchi et al., 1992 and Croisy, 1994; Conolly and Edmonds, 1994).

4. CONCLUSION

The aim of this paper has been to provide a brief summary of the state of the art of models of architectures for intelligent HMIs. For this purpose we have first presented aspects which are primarily of interest to both cognitive and computer science, and then concepts related to interfaces which rightly or wrongly are called “intelligent”, for which purpose we have used the ARCH model as a frame of reference.

Five types of intelligent interfaces have been reviewed. We have described their intelligence along two main aspects (see Figure 14):

- the knowledge that the interface comprises about the user and his/her tasks,
- the autonomy that the interface can reach in the management of the human-machine interaction.

In fact, these aspects are correlated because the autonomy relies strongly on the knowledge possessed by the interface. Obviously, it requires the capacity to use this knowledge but this is more a technical question than a conceptual one. The position exposed in this paper is clearly a user-centred approach that uses the models developed in cognitive ergonomics, rather than a technology-driven approach. This is the point of view of “what we advocate” more than a review of what is realised in most daily developments.

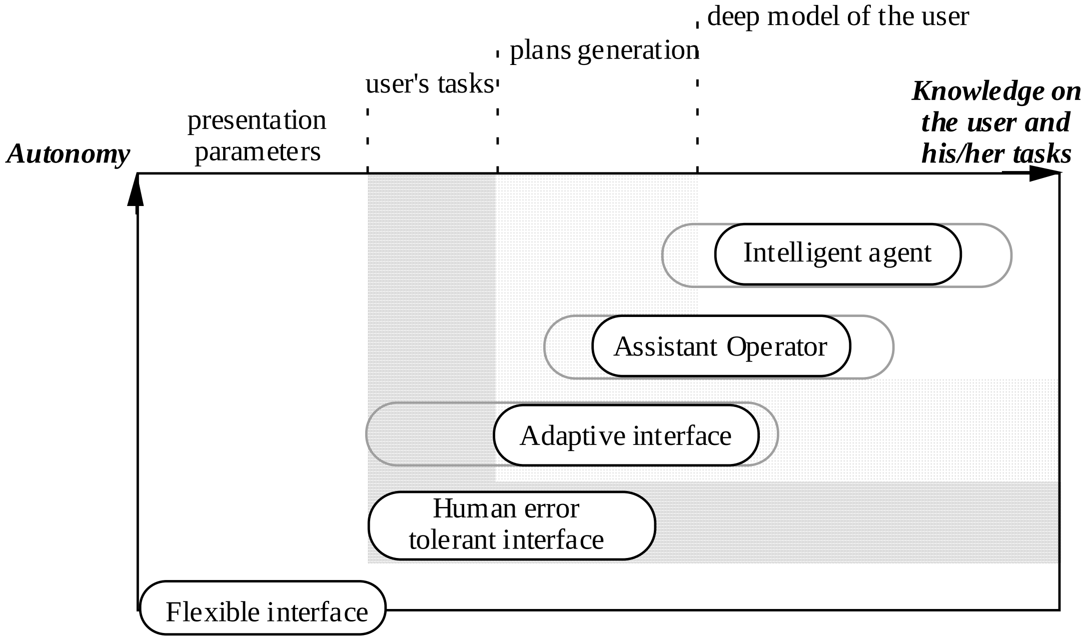

Figure 14. Intelligent interfaces typology

This typology of the intelligent interfaces makes distinctions that are based on their functional characteristics, ie. what they are able to achieve. The way these capacities are implemented is another and independent problem, for which there are many architectural solutions. Some of these solutions have been given as examples. They have been exposed via the ARCH model, which was chosen for its basic, generic and adaptable features. We have clearly taken a stand at the modelling level; the design of the interface architecture is a different subject of study, which has not been deepened here.

In view of the different approaches that have been examined, it is obvious that the development of interfaces requires an interdisciplinary approach because of the variety of problems, whether technical or human, which have to be considered. In addition to purely technical aspects of development (computer science, software and hardware engineering), the development of intelligent interfaces involves methods and models of classical artificial intelligence, as well as, more and more, models from distributed artificial intelligence. Next one must cite the need for cognitive science models, and cognitive models of tasks and of users, the ideas which come from cognitive ergonomics and cognitive and work psychology, and even, indeed research into human error and human reliability. There are even aspects of sociology which are relevant to cooperative work and organisational aspects of work, and these are appearing more and more in this domain. It seems obvious that different models and different tools will be needed as a function of the domain of application, such as tutoring systems, office automation, domestic applications, industrial process control, internet (information research, electronic commerce...), manufacturing, etc.

It is only some fifteen years since research in this area began, and even though intelligent interfaces have hardly emerged from the egg, several major lines of development can be seen.

A first research line consists of the improvement of the models, and the knowledge in general, used by the interface to determine its behaviour:

- There is a need to discover how to model tasks so that they provide a database, which an intelligent interface can use to deduce the needs of the users. Are the most famous methods well adapted? Do we need specific task models?
- In line with that need, and following Maes (1994) and Herrmann (1996), there is a need to study how automatic symbolic learning can be used to develop and update knowledge bases for intelligent HMIs. Indeed many Machine Learning methods exist (Michalski and Kodratoff, 1990; Buchanan and Wilkins, 1993), what are the methods which are really useable?
- Starting from existing models of cognition, we need to produce better generic models of users of interactive systems. Often issued from cognitive sciences, could these models become, more or less rapidly, engineering tools for computer scientists and automatists?
- Since there seems to be many domains of application with highly specific characteristics and demands, these will need to be studied individually before intelligent interfaces can be developed for them.
- The idea of levels of autonomy for intelligent interfaces needs to be further examined, and related to different criteria, such as performance, human preferences, quality of production,

safety, support, etc. It seems obvious that multi-criterion decision-making needs to be introduced. [10] Indeed, the importance given to each criterion can vary largely from one application field to another, and also from one situation (normal or abnormal) to another, according to the dynamics of the application.

A second major line concerns the methods used to develop intelligent interfaces: analysis and design methods are needed to help in the development of all the approaches reviewed in this paper. For instance, there are many software engineering methods, such as SASD (Structured Analysis and Structured Design; deMarco, 1979), SA-RT (Structured Analysis - Real Time; Hatley and Pirbhai, 1988), OMT (Object Modeling Technique; Rumbaugh et al., 1991), MERISE (Tardieu et al., 1985; 1991), SSADM (Longworth and Nicholls; 1986), and so on. Many methods are general (like OMT), or dedicated to specific application fields (such as the real time systems for SA-RT, or the information systems for MERISE). Could such methods be adapted and take into account intelligent interfaces concepts? Is it necessary to propose specific methods? A study could be very useful for that purpose. Moreover just as there are now development environments and shells for developing systems which use knowledge bases, and environments and shells for developing HMIs, we need the same facilities - in form of Computer Assisted Software Engineering (CASE) tools, for the development of “intelligent” HMIs. Such CASE tools would support special analysis and design methods.

And finally, we see a third research line in the extension of the intelligent interface notion to a larger area. Specifically, it needs to be considered in relation to the development of intelligent HMIs for cooperative work and for multimodal interaction. As regards the area of cooperative work, the agent notion can be useful because it is well-adapted to modelling groups of interacting humans and computers. In this way, the interaction stays no more in the “couple” composed of one human and one computer but in the group including human agents and computer agents. That introduces specific cognitive and design problems (some research directions are given in Le Strugeon et al., 1996). The use of multimodal tools is another way to widen and enrich the interaction (Earnshaw and Vince, 1995). The interface has different tools at its disposal to dialogue with the user. Obviously, each of these tools has its own characteristics and must be used in the situations to which it is adapted. In other words, the intelligent interface must decide which kind of media is the most adapted in the current situation, and the user(s) concerned.

5. ACKNOWLEDGMENTS

The authors thank sincerely the anonymous referees of the journal for their numerous helpful comments. They thank also Joe GALWAY and the Professor Neville MORAY for considerable assistance in the translation of the article.

---

[10] See the Journal of Multi-criteria Decision Analysis (Wiley).

REFERENCES

Abed, M., & Angue, J.C. (1994). A new method for conception, realisation and evaluation of man-machine systems. *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Texas, October.

Amalberti, R., De Montmollin, M., & Theureau, J. (Eds.) (1991). *Modèles en analyse du travail*. Liège : Mardaga.

Amalberti, R., & Deblon, F. (1992). Cognitive modeling of fighter aircraft process control : a step towards an intelligent onboard assistance system. *International Journal of Man-Machine Studies*, 36 (5).

Anderson, J.R. (1983). *The architecture of cognition*. Cambridge : Harvard University Press.

Anderson, J.R. (1993). *Rules of the mind*. Hillsdale, NJ : Lawrence Erlbaum.

The UIMS Developers Workshop (1992). A metamodel for the runtime architecture of an interactive system. *SIGCHI Bulletin,* volume 24, numero 1, pp. 32-37.

Ashby, W.R. (1956). *An introduction to cybernetics*. London: Methuen & Co.

Avouris, N.M., & Liedekerke, M.H. (1993). User interfaces design for cooperating agents in industrial process supervision and control applications. *International Journal of Man-Machine Studies*, 38, pp. 873-890.

Baddeley, A.D., & Hitch G. (1974). Working memory. In G.H. Bower (Ed.), *The Psychology of Learning and Motivation*, Academic Press.

Balbo, S. (1994). *Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation*. Ph.D Thesis, Université de Grenoble I.

Banks, S.B., Harrington, R.A., Santos, E., & Brown, M. (1997). Usability testing of an intelligent interface agent. *Proceedings INTERFACES'97*, EC2 Press, LLIA N° 123, pp.121-123, May.

Barthet, M.F. (1995). The DIANE Method and its connection with MERISE Method. *Proceedings IEA World Conference "Ergonomic design, interfaces, products, Information"*, October 16-20, Rio de Janeiro, Brazil, pp. 106-110.

Barnard, P.J. (1987). Cognitive Ressources and the learning of Human-Computer Dialogs. In *Interfacing Thought,* Caroll J.M. (Ed.), MIT Press, pp. 112-158.

Barnard, P.J., & Teadsale, J. (1991). Interacting Cognitive Subsystems: A systemic approach to cognitive affective interaction and chjange. *Cognition and Emotion,* 5 (1), pp. 1-39.

Bass, L. Little, R., Pellegrino, R., Reed, S., Seacord, S., Sheppard, S., & Szesur, M. (1991). The Arch model : Seeheim revisited. *Proceedings of User Interface Developers' Workshop*, Seeheim, April.

Beka Be Nguema, M., Waroux, D., & Malvache, N. (1992). Contributing to the design of a human error tolerant man-machine interface during the control of a simple simulated industrial process. *Proceedings 11th European Annual Conference on Human Decision Making and Manual Control*, 17-19 novembre, Valenciennes, France.

Benysh, D.V., Koubek, R.J., & Calvez, V. (1993). A comparative review of knowledge structure measurement techniques for interface design. *International Journal of Human-Computer Interaction*, 5 (3), pp. 211-237.

Bersini, H. (1989). *La modélisation cognitive dans l'étude des systèmes homme-machines*. Ph.D Thesis, université libre de Bruxelles, Belgium.

Bisson, P., Soulard, P., Raimondo, A., & Aknin, P. (1992). Interfaces homme-machine multimodales et adaptatives pour systèmes opérationnels. *Proceedings congrès "L'interface des mondes réels et virtuels"*, 23-27 mars, Montpellier, EC2, pp. 591-603.

Boy, G. (1983). *Etude de la charge de travail dans une cabine de pilotage. Mode opératoire et stratégie de récupération dans le système MESSAGE*. Rapport CERT DERA/AIRBUS Industrie, n. 11/7275, August.

Boy, G. (1986). An expert system for fault diagnosis in orbital refueling operations. *Proceedings 24th Aerospace Science Meeting*, AIAA'86, Reno, Nevada, January.

Boy, G. (1988). Operator assistant systems. In *Cognitive engineering in complex dynamic worlds*, edited by E. Hollnagel, G. Mancini, D.D. Woods, Academic Press.

Boy, G., & Tessier, C. (1985). Cockpit analysis and assessment by the MESSAGE methodology. *Proceedings 2nd IFAC Conference on "Analysis and Evaluation of man-machine systems"*, Varese, Italy, pp. 46-56, September.

Boy, G. (1989). The block representation in knowledge acquisition for computer integrated documentation. *Proceedings 4th Knowledge acquisition for knowledge-based systems AAAI Workshop*, Banff, Canada, October 1-6.

Boy, G. (1990). Advanced interaction media. *Proceedings International Conference Human Machine Interaction and Artificial Intelligence in Aeronautics and Space*, Odyssud, Toulouse-Blagnac, 26-28 september.

Boy, G. (19991). *Intelligent Assistant Systems*. New York: Academic Press.

Brajnik, G., Guida, G., & Tasso, C. (1990). User modelling in expert man-machine interface : a case study in intelligent information retrieval. *IEEE Transactions on Systems, Man and Cybernetics*, 20 (1), pp. 167-185.

Buchanan, B.G., & Wilkins, D.C. (1993). readings in knowledge acquisition and learning: automating the construction and improvement of expert systems. San Mateo, CA: Morgan Kaufmann.

Bullinger, H.J., & Faehnrich, K.P. (1984). Symbiotic man-computer interfaces and the user assistent concept. *Proceedings of the First USA-Japan Conference on Human-Computer Interaction*, Honolulu, 1984.

Cacciabue, P.C., Decortis, F., & Drozdowicz, B. (1990). Modélisation cognitive de l'activité d'un opérateur contrôlant un système complexe. *Proceedings Symposium sur la Psychologie du Travail et les Nouvelles Technologies*, Université de Liège, 17-18, May.

Cacciabue, P.C., Mancini, G., & Bersini, U. (1990). A model of operator behaviour for man-machine system simulation. *Automatica*, 26 (6), pp. 1025-1034.

Communications of the ACM (1994). *Special issue : Intelligent Agents*. 37 (7), ACM Press, july.

Conant, R.C., & Ashby, W.R. (1970). Every good regulator of a system must be a model of that system. *International Journal of System Science*, 1/2, pp. 89-97.

Card, S.K., Moran, T.P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ : Erlbaum.

Card, S.K. (1989). Human factors and artificial intelligence. In P.A. Hancock and M.H. Chignell (Eds.), *Intelligent interfaces : theroy, research and design*, Amsterdam, North Holland.

Carter, J.A. (1991). Combining task analysis with software engineering in a methodology for designing interactive systems. In *Taking Software Design Seriously*, J. Karat (Ed.), Academic Press.

Chaib-Draa, B., Moulin, B., Mandiau, R., & Millot, P. (1992). Trends in Distributed Artificial Intelligence. *AI Review*, Yazdani (Ed.), pp. 27-42, 1992.

Chignell, M.H., & Hancock, P.A. (1988). Intelligent interface design. In Helander, M. (Ed.), *Hanbook of Human-Computer Interaction*, Elsevier Science Publishers B.V. (North Holland), pp. 969-995.

Chignell, M.H., Hancock, P.A., & Loewenthal A. (1989). An introduction to intelligent interfaces. In P.A. Hancock and M.H. Chignell (Eds.), *Intelligent interfaces : theroy, research and design*, North Holland.

Connoly, J.H., & Edmonds, E.A. (Eds.) (1994). *CSCW and artificial intelligence*. Springer-Verlag, London.

Cooke, N.J. (1994). Varieties of knowledge elicitation techniques. *International Journal of Human-Computer Studies*, 41, pp. 801-849.

Coutaz, J. (1987). PAC, an implementation model for dialog design. *Proceedings Interact'87 Conference*, H.J. Bullinger, B. Shackel (Eds.), North Holland, September, pp. 431-436.

Coutaz, J. (1990). *Interfaces homme-ordinateur : conception et réalisation*. BORDAS, Paris.

Coutaz, J., Crowley, J., & Laugier, C. (1992). Intelligence artificielle et interfaces homme-machine. *Proceedings 4èmes journées sur l'intelligence artificielle*, Marseille, pp. 169-186, October.

Croisy, P. (1994). Modèle multi-agent et conception d'applications coopératives interactives. *Proceedings Journées IHM'94*, Lille, 8-9, décember, pp. 139-144.

Dance, J.R., Granor, T.E., Hill, R.D., Hudson, S.E., Meads, J., Myers, B.A., & Shulert, A. (1987). The run-time structure of UIMS-supported applications. *Computer Graphics*, 21 (2), april.

DeMarco, T. (1979). *Structured analysis and system specification*. Yourdon Press, NY, Englewood Cliffs, Prentice Hall.

Dewan, P. (1992). Principle of designing multi-user interface development environments. In *Engineering for Human-Computer Interaction*, J.J. Larson and C. Unger (Eds.), Elsevier Science Publishers B.V. (North-Holland), IFIP, pp. 35-50.

Diaper, D. (1989). *Task analysis for human-computer interaction*. Ellis Horwood Limited, Chichester, United Kingdom.

Dieng, R. (1993). Méthodes et outils d'acquisition des connaissances. In *L'ergonomie dans la conception des Projets Informatiques*, J.C. sperandio (Ed.), Octares Editions, Toulouse, pp. 335-411.

Duncan, K.D. (1981). Training for fault diagnosis in industrial process plant. In *Human Detection and Diagnosis of System Failures*, J. Rasmussen and B. Rouse (Eds.), Plenum Press, New-York.

Duval, T. (1994). Modèles d'architecture pour les applications graphiques interactives : la famille Seeheim. *Revue de CAO et d'infographie*, volume 9, numéro 4, pp. 529-555.

Earnshaw, R.A., & Vince, J.A. (Eds.) (1995). Multimedia systems and applications. London, Academic Press.

Edmonds, E.A. (1981). Adaptative Man-Computer Interfaces. In : M.J. Coombs and J.L. Alty (eds.), *Computing skills and the user interface*, London, Academic press.

Edmonds, E.A. (Ed.) (1992). *The separable user interface*. Academic Press.

El Farouki, L., Scapin, D., & Sebillotte, S. (1991). *Prise en compte des tâches du contrôleur pour l'ergonomie des interfaces*. Final Report CENA/N.91223, Convention entre le CENA et l'INRIA, August.

Falzon, P. (1989). *Ergonomie cognitive du dialogue*. Editions P.U.G., Grenoble.

Ferrarri, S. (1994). Modèle de la tâche dans le cadre d'une interface de dialogue multimodale. *Proceedings Journées IHM'94*, Lille, 8-9 décembre, pp. 49-54.

Foley, J.D., & Van Dam, A. (1982). *Fundamentals of interactive computer graphics*. Addison-Wesley.

Fournier, J.P. (1994). Vers des programmes dociles. *Techniques et Sciences informatiques*, 13 (1), pp. 79-104.

Furtado, E., Santoni, C., & François, P. (1995). Adaptive human computer interfaces for supervision systems. In *Symbiosis of Human and Artifact*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Elsevier Science B.V., pp. 1077-1082.

Gasser, L., Huns, M.N. (Eds.) (1989). *Distributed artificial intelligence, vol. 2*. Pitman, London.

Gasser, L. (1991). Social conceptions of knowledge and action : DAI foundations and open systems semantics. *Artificial Intelligence*, vol. 47, pp. 107-138.

Gerlach, M., Onken, R., Prevot, T., & Ruckdeschel, W. (1995). The cockpit assistant system CASSY, design and in-flight evaluation. In *Symbiosis of Human and Artifact*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Elsevier Science B.V., pp. 759-764.

Gibson, J.J. (1979). *The ecological approach to visual perception*. Boston:Houghton-Mifflin.

Goldberg, A. (1984). *Smalltalk-80, the interactive programming environment*. Addison-Wesley.

Gray, W.D., John, B.E., & Artwood, M.E. (1992). The precis of project Ernestine or an overview of a validation of GOMS. *Proceedings CHI'92 Conference*, P. Bauersfeld, J. Bennett and G. Lynch (Eds.), ACM, Addison-Wesley.

Grislin, M., & Kolski, C. (1996). Evaluation des interfaces homme-machine lors du développement des systèmes interactifs. *Technique et Science Informatiques*, 15 (3), pp. 265-296.

Grislin, M., Kolski, C., & Angué, J.C. (1995). Human-computer interface evaluation in industrial commplex systems : a review of usable techniques. *Proceedings 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems*, M.I.T., Cambridge, June 27-29.

Hancock, P.A., & Chignell, M.H. (Eds.) (1989). *Intelligent interfaces : theroy, research and design*. Amsterdam, North Holland.

Hartson, H.R., & Gray, P.D. (1992). Temporal aspects of tasks in the user action notation. *Human-Computer Interaction*, vol. 7, pp. 1-45.

Hatley, D. & Pirbhai, I. (1988). *Strategies for real-time systems specification*. Prentice Hall, N.Y.

Hefley, W.E. (1990). Architecture for adaptable Human-Machine Interface. *Proceedings second International Conference on "Human aspects of advanced manufacturing and hybrid automation"*, Honolulu, Hawaii, August 12-16.

Herrmann, J. (1996). Different ways to support intelligent assistant systems by use of machine learning methods. *International Journal of Human-Computer Interaction*, 8 (3), pp. 287-308.

Hill, R.D. (1992). The abstraction-link-view paradigm:using constraints to connect user interfaces to applications. *Proceedings CHI'92 Conference*, Monterey, California, May 3-7, ACM Press.

Hoc, J.M., & Amalberti, R. (1995). Diagnosis : some theoretical questions raised by applied research. *Current Psychology of Cognition*, 14 (1), pp. 73-101.

Hollnagel, E. (1989). The design of fault tolerant systems : prevention is better than cure. *Proceedings 2nd European Meeting on cognitive science approaches to process control*, October 24-27, Siena, Italy.

Hollnagel, E. (1994). *Human reliability analysis, context and control*. Academic Press, London.

Irving, S., Polson, P., & Irving, J.E. (1994). A GOMS Analysis of the advanced automated cockpit. *Proceedings CHI'92 Conference,* B. Adelson, S. Dumais and J. Olson (Eds.), ACM Press, Addison-Wesley.

John, B. (1995). Why GOMS. *Interactions*, volume 11(4), pp. 80-89, October.

John, B.E., & Vera, A.H. (1992). A GOMS analysis of a graphic, machine-paced, highly interactive task. *Proceedings CHI'92 Conference,* P. Bauersfeld, J. Bennett and G. Lynch (Eds.), ACM Press, Addison-Wesley.

Jurain, T. (1991). De l'écrit à l'écran, étude et classification des aides logicielles au développement d'interfaces graphiques. *Génie logiciel et systèmes experts*, 24, september.

Karsenty, S., & Weikart, C. (1991). Une extension de l'interface d'application dans le modèle de Seeheim. *Proceedings IHM'91,* Dourdan, décember.

Kieras, D.E. (1988). Towards a practical GOMS model methodology for user interface design. In *The Handbook of Human-Computer Interaction*, Helander, M. (Ed.), Amsterdam: North-Holland.

Kirwan, B., & Ainsworth, L.K. (1992) *A guide to task analysis*. London: Taylor and Francis.

Klein, G.A. (1989). Recognition-Primed Decisions. In Rouse, W.B. (Ed.), *Advances in Man-Machine Systems Research* (Vol. 5), pp. 47-92, Greenwich, CT: JAI Press.

Klein, G.A. (1993). A Recognition-Primed Decisions (RPD) model of rapid descision making. In Klein, G.A., Orasanu, J., Calderwood, R. & Zsambok, E. (Eds.), *Decision making in action: models and methods,* pp. 138-147, Ablex Publishing Corporation, Norwood, NJ.

Kolodner, J.L. (1993). *Case-based reasoning*. Morgan Kaufman, San Mareo, CA.

Kolski, C. (1997). Interfaces homme-machine, application aux systèmes industriels complexes (second edition). Editions Hermès, Paris, january.

Kolski, C., Le Strugeon, E., & Tendjaoui, M. (1993). Implementation of AI techniques for "intelligent" interface development. *Engineering Application of Artificial Intelligence*, vol. 6 (4), pp. 295-305.

Kolski, C., Tendjaoui, M., & Millot, P. (1992). A process method for the design of "intelligent" man-machine interfaces : case study : "the Decisional Module of Imagery". *International Journal of Human Factors in Manufacturing,* vol. 2 (2), pp. 155-175.

Leontiev, A. (1974). The problem of activity in psychology. *Soviet Psychology,* 13, pp. 4-33.

Leplat, J. (1985). *Erreur humaine, fiabilité humaine dans le travail*. Armand Colin, Paris.

Le Strugeon, E., Grislin, M., & Millot, P. (1995). Toward the application of multiagent techniques to the design of human-machine systems organizations. *Proceedings 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems,* M.I.T., Cambridge, June 27-29.

Le Strugeon, E., Kolski, C., Mandiau, R., & Tendjaoui, M. (1996). Intelligent agents concepts for computer-supported collaborative work systems in complex industrial systems - Research directions. *Proceedings 2nd International Conference on the Design of Cooperative Systems*, Juan-les-Pins, France, June, 12-14.

Lim, K.Y., Long, J.B., & Silcock, N. (1992). Instanciation of task analysis in a structured method for user interface design. *Proccedings of the 11th Interdisciplinary Workshop on Informatics and Psychology : Task Analysis in Human-Computer Interaction*, Scarding, June, 9-11.

Lindsay, P.H., & Norman, D.A. (1980). *Traitement de l'information et comportement humain, une introduction à la psychologie*. Editions Etudes Vivantes, Montréal-Paris.

Longworth, G., & Nicholls, D. (1986). *SSADM Manuel, volume 1 : tasks and terms, volume 2 : technique and documentation*. NCC, The National Center for Information Technology, vers. 3.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37 (7), July, pp. 31-40.

Maes, P., & Kozierok, R. (1993). Learning interface agents. *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington D.C., AAAI Press.

Mandiau, R., & Le Strugeon, E. (1995). Explicit Representation of a Multiagent Group: the Shared Plan. *Proceedings 7th International Conference on Systems, Man and Cybernetics*, Vancouver, Canada, October.

Mandiau, R., Kolski, C., Millot, P., & Chaib-Draa, B. (1991a). A new approach for the cooperation between human(s) and assistance system(s), a system based on intentional states. *Proceedings World Congress on Expert Systems*, Orlando, Florida, Dec. 16-19.

Mandiau, R., Kolski, C., & Millot, P., (1991b). Perspective of human-machine system modelling by logical approach : contribution of technics issued from distributed problem solving. *Proceedings 11th Congress of the International Ergonomics Association*, Paris, July 15-20.

Masson, M. (1994). *Prévention automatique des erreurs de routine*. Ph.D. Thesis in Psychology, Université de Liège, Belgium, July.

McGrew, J. (1991). Tools for task analysis : graphs and matrices. In *Taking Software Design Seriously*, J. Karat (Ed.), Academic Press.

Michalski, R.S., & Kodratoff, Y. (1990). Research in Machine Learning, recent progress, classification of methods and future directions. In *Machine Learning : An artificial Intelligence Approach*, vol. 3, Y. Kodratoff and R.S. Michalski (Eds.), Palo Alto, California.

Miller, G.A. (1975). *The psychology of communication. Basic Books*, New York, 2nd Ed.

Millot, P. (1988). *Supervision des procédés automatisés et ergonomie*. Editions Hermes, Paris, Décembre.

Millot, P. & Hoc, J.M. (1997). Human-machine cooperation: metaphor or possible reality? *Proceedings European Conference on Cognitive Science*, Manchester, United Kingdom, April, 9-11.

Muller, M.J., Carr, R., Ashworth, C., Diekmann, B., Wharton, C., Eickstaedt & Clonts, J. (1995). Telephone operators as knowledge workers: consultants who meet customer needs. *CHI'95 Conference Proceedings*, I.R. Katz, R. Mack and L. Marks (Eds.), ACM Press, Addison-Wesley, pp. 130-137.

Myers, B.A. (1988). *Creating user interfaces by demonstration*. Academic Press.

Myers, B.A., McDaniel, R.G., & Kosbie D.S. (1993). Marquise: creating complete user interfaces by demonstration. *Proceedings INTERCHI'93 Human Factors in Computing Systems*, ACM Press, Addison-Wesley, pp. 293-300.

Moray, N. (1986). Monitoring behavior and supervisory control. In K.R. Boff, L. Kaufman, and J.P. Thomas (Eds), *Handbook of Perception and Human Performance*, chapter 45, New York:Wiley.

Moray, N., Jones, B.G., Sanderson, P.M., Reising, D.V., Shaheen, S., & Rasmussen, J. (1995). The "Bird's Foot" integrated graphical interface for NPP operation. In *Symbiosis of Human and Artifact, Social Aspects of Human-Computer Interaction*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Elsevier Science B.V., pp. 995-998.

Moran, T.P. (1981). The Command Language Grammar : a representation of the user interface of interactive computer system. *International Journal of Man-Machine Studies*, 15, pp. 3-50.

Muller, M.J. (1988). Disentangling application and presentation in distributed computing : architecture and protocole to enable a flexible common user interface. *IEEE Transactions on Systems, Man and Cybernetics*, 18 (4), july/August.

Nakauchi, Y., Itoh, Y., Sato, M., & Anzai, Y. (1992). Modelling and implementation of multi-agent interface system for computer-supported co-operative work. *Ergonomics*, 35 (5/6), pp. 565-576.

Nanard, J., & Nanard, M. (1995). Hypertext design environments and the hypertext design process. *Communications of the ACM*, 38 (8), August, pp. 49-56.

Nardi, B.A. (Ed.) (1995). *Context and consciousness, Activity Theory and human-Computer Interaction*. The MIT Press.

Newell, A., & Simon, H.A. (1972). *Human problem solving*. Prentice Hall, Englewood Cliffs, HJ.

Newell, A. (1990). *Unified Theories of Cognition*. The William James Lectures. Cambridge, MA: Harvard University Press.

Nigay, L., & Coutaz, J. (1991). Building user interfaces: organizing software agents. In *Proceedings ESPRIT'91 Conference,* Bruxelles, november, pp. 707-719.

Nigay, L., & Coutaz, J. (1995). A generic platform for adressing the multimodal challenge. *Proceedings CHI'95 Conference*, I.R. Katz, R. Mack and L. Marks (Eds.), ACM Press, Addison-Wesley, pp. 98-105.

Nii, H.P. (1986a). Blackboard systems : blackboard application systems, blackboard systems from a knowledge engineering perspective. *AI Magazine*, 7 (2), pp. 82-106.

Nii, H.P. (1986b). Blackboard systems : the blackboard model of problem-solving and the evolution of blackboard architectures. *AI Magazine,* 7 (3), pp. 39-53.

Norman, D.A. (1986). Cognitive engineering. In D.A. Norman & S.W Draper (Eds), *User centred system design : new perspectives on human computer interaction*. Hillsdale, NJ : Erlbaum.

Norcio, A.F., & Stanley, J. (1989). Adaptive human-computer interface : a literature survey and perspective. *IEEE Transactions on Systems, Man and Cybernetics*, 19 (2), March/April.

Pfaff, G.E. (1985). *User interface management system*. Springer-Verlag.

Payne, S.J., & Green, T.R.G. (1989). Task-Action Grammar: the model and developments. In *Task analysis for HCI*, Diaper (Ed), John Wiley and Sons, pp. 75-107.

Rasmussen, J. (1980). The human as a system component. In H.T. Smith and T.R.G. Green (Eds.), *Human Interaction with Computer*, London Academic Press.

Rasmussen, J. (1981). Models of mental strategies in process plant diagnosis. In Rasmussen, J. & Rouse, W.B. (Eds.), *Human detection and diagnosis of system failures,* pp. 241-258, Plenum Press, New-York.

Rasmussen, J. (1983). Skills, rules and knowledge; signals, signs and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics,* volume 13, n°3, pp. 257-266.

Rasmussen, J. (1984). Strategies for state identification and diagnosis in supervisory control tasks, and design of computer-based support systems. *Advances in Man-Machine Systems Research*, 1, pp. 139-193.

Reason, J. (1987). A framework for classifyng errors. In *New Technology and Human Error*, J. Rasmussen and J. Leplat (Eds.), Viley and Sons, Chichester.

Reason, J. (1990). *Human Error*. Cambridge : Cambridge University Press.

Reisner, P. (1981). Formal grammar and human factors design of an interactive graphics system. *IEEE Transactions on Software Engineering,* 7 (2), March.

Reisner, P. (1984). Formal Grammar as a tool for analyzing ease of use, some fundamental concepts. In *Human Factors in Computer Systems,* J.C. Thomas, M.L. Schneider (Eds.), Ablex, New Jersey.

Rouse, W.B. (1988). Adaptive aiding for Human-Computer control. *Human Factors,* 30 (4), pp. 431-443.

Rouse, W.B., & Morris, N.M. (1985). Conceptual design of an human error tolerant interface for complex engineering systems. *Proceedings 2nd IFAC Conference on "Analysis and Evaluation of man-machine systems"*, Varese, Italy, pp. 46-56, Septembre.

Rubin, K., Jones, P.M., & Mitchell, C.M. (1988). OFMspert : inference of operator intentions in supervisory control using blackboard architecture. *IEEE Transactions on Systems, Man and cybernetics*, 18 (4), pp. 618-636, july/august.

Rumbaugh, J. Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). *Object-oriented modeling and design*. Prentice-Hall.

Sacerdotti, E.D. (1977). *A structure for plans and behavior*. Elsevier, computer science libary.

Sakuma, A., Itoh, J., Yoshikawa, E., & Monta, K. (1995). Simulation study of an ecological interface for nuclear power plants. In *Symbiosis of Human and Artifact, Social Aspects of Human-Computer Interaction*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Elsevier Science B.V., pp. 989-994.

Schneider-Hufschmidt, M., Kühme, T., & Malinkowski, U. (Eds.) (1993). *Adaptive User Interfaces*. North Holland.

Sebillotte, S., & Scapin, D. (1994). From users' task knowledge to high level interface specification. *International Journal of Human-Computer Interaction*, 6 (1).

Senach, B. (1990). Evaluation de l'ergonomie des IHM. Proceedings Congrès ERGO-IA'90, ergonomie et informatique avancée, Biarritz, France.

Shalin, V.L., Levi, K.R., & Perschbacher, D.L. (1990). Learning pilot tactics for an intelligent pilot's assistant. *Proceedings International Conference Human Machine Interaction and Artificial Intelligence in Aeronautics and Space*, Odyssud, Toulouse-Blagnac, 26-28 septembre.

Sheridan, T.B., & Verplanck, W. L. (1978). *Human and computer control of undersea teleoperators*. Technical report, Man-Machine Systems Laboratory, Department of mechanical Engineering, Massachuisetts Institute of Technology, Cambridge. Mass.

Sperandio, J.C. (1991). Les méthodes d'analyse du travail en psychologie ergonomique. In *La recherche en psychologie (domaines et méthodes)*, J.P. Rossi (Ed.), Bordas, Paris.

Tarby, J.C. (1993). *Gestion automatique du dialogue homme-machine à partir des spécifications conceptuelles*. Ph.D. Thesis, Toulouse, september.

Tardieu, H., Rochfeld, O., & Colleti, R. (1991). *La méthode Merise, principes et outils, 2ème édition*. Editions d'Organisation (tome 1), Paris.

Tardieu, H., Rochfeld, O., Colletti, R., Panet, G., & Vahee, G. (1985). *La méthode Merise, démarche et pratiques*. Editions d'Organisation (tome 2), Paris.

Tauber, M. (1990). ETAG: Extended Task Action Grammar - A language for the description of the user’s task language. In *Human-Computer Interaction - INTERACT'90*, Diaper et al. (Eds.), p. 163-168, North-Holland.

Tendjaoui, M., Kolski, C., & Millot, P. (1991). An approach towards the design of intelligent man-machine interfaces used in process control. *International Journal of Industrial Ergonomics*, 8, pp. 345-361.

Theureau, J. (1992). *Le cours d'action : analyse sémio-logique*. Berne, Peter Lang.

Vicente, K.J., & Rasmussen, J. (1992). Ecological interface design: theoretical foundations. *IEEE Transactions on Systems, Man and Cybernetics*, 22, pp. 589-606.

Vicente, K.J. (1995). Ecological interface design: a research overview. *Proceedings 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems*, M.I.T., Cambridge, June 27-29.

Waern, Y. (1989). *Cognitive aspects of computer supported tasks*. John Wiley & Sons Ltd.

Watanabe, O., Takaura, K., Fujita, Y., & Hayashi, Y. (1995). Evaluation of ecological interface design. In *Symbiosis of Human and Artifact, Social Aspects of Human-Computer Interaction*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Elsevier Science B.V., pp. 977-982, 1995.

Weiner, A.J., Mitchell, C.M. (1995). FIXIT: a case-based architecture for computationally encoding fault management systems. In Sheridan, T.B. (Ed.), *Analysis, Design and Evaluation of Man-Machine Systems*, pp. 159-164, Oxford: Elsevier science.

Williges, R.C., Williges, B.H., & Elkerton, J. (1987). Software Interface Design. In G. Salvendy (ed.), *Handbook of Human factors*, pp. 1416-1449, New york, Wiley.

Wilson, J.R., & Corlett, E.N. (Eds.) (1990). *Evaluation of human works, a practical ergonomics methodology*. Taylor & Francis.

Wittig, T. (Ed.) (1992). *ARCHON : an architecture for multi-agent system*. Chichester : Ellis Horwood.

Woods, D.D., & Roth, E.M. (1987). *Models of cognitive behaviour in nuclear power plant personnel*. Report NUREG/CR-4532.

Yoshida, K., Yokobayashi, M., Kawase, K., & Tanabe, F. (1995). Computer simulation system of cognitive man-machine interaction in accidental situation of nuclear power plant. In *Symbiosis of Human and Artifact*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Elsevier Science B.V., pp. 939-944.

Young, R.M., & Whittington, J. (1990). A knowledge analysis of interactivity. *Proceedings INTERACT'90*, Diaper D., Cockton G. and Shackel B. (Eds.), Elsevier Scientific Publishers B.V., pp. 207-212, Cambridge (UK), 27-31 August.

A version of this paper has been published in: *International Journal of Human-Computer Interaction*, 10 (3), pp. 193-231, 1998.

Christophe Kolski is Professor in computer science at the University of Valenciennes where he teaches artificial intelligence, software engineering and human-machine communication. His research is concerned with the design and evaluation of HMI, so-called “intelligent” interfaces, and user aiding. He is the author of a book entitled "Human-Machine Interfaces, application to industrial complex systems (in French)", second edition, published by Editions HERMES (Paris) in 1997.

Emmanuelle Le Strugeon is Assistant professor in computer science at the University of Valenciennes, where she teaches computer engineering and data warehousing. She received a Ph. D. degree in 1995 for work on artificial intelligence and more particularly on the organisation of multi-agent systems. Her research interests include humans-computers interactions, so-called “intelligent” interfaces, agent modelling and collaborative work.