HUMAN-CENTERED AUTOMATION: A MATTER OF AGENT DESIGN AND COGNITIVE FUNCTION ALLOCATION

Guy Boy
European Institute of Cognitive Sciences and Engineering (EURISCO International), 4 Avenue Edouard Belin, 31400 Toulouse, France.

Abstract: This chapter presents an analytical framework that brings answers to and overcomes the "classical" debate on direct manipulation versus interface agents. Direct manipulation is always appropriate when the system to be controlled is simple. However, when users need to interact with complex systems, direct manipulation is also complex and requires a sufficient level of expertise. Users need to be trained, and in some cases deeply trained. They also need to be assisted to fulfill overall criteria such as safety, comfort or high performance. Artificial agents are developed to assist users in the control of complex systems. They are usually developed to simplify work, in reality they tend to change the nature of work. They do not remove training. Artificial agents are evolving very rapidly, and incrementally create new practices. An artificial agent is associated to a cognitive function. Cognitive function analysis enables human-centered design of artificial agents by providing answers to questions such as: Artificial agents for what? Why are artificial agents not accepted or usable by users? An example is provided, analyzed and evaluated. Current critical issues are discussed.

Key words agents; cognitive functions; human-centered automation; safety; direct manipulation; expertise.

1. INTRODUCTION

The concept of artificial agent, and automaton in general, is very clumsy. The term clumsy automation was introduced by Earl Wiener who has studied aircraft cockpit automation for the past three decades (Wiener, 1989). Wiener criticizes the fact that the traditional answer of engineers to human-machine interaction problems was to automate. His research results suggest that particular attention should be paid to the way automation is done. In addition to these very well documented results, there are pessimistic views on the development of software agents:

"Agents are the work of lazy programmers. Writing a good user-interface for a complicated task, like finding and filtering a ton of information, is much harder to do than making an intelligent agent. From a user's point of view, an agent is something you give slack to by making your mind mushy, while a user interface is a tool that you use, and you can tell whether you are using a good tool or not." (Lanier, 1995, page 68).

This is a partial view of the problem. Agents cannot be thrown to the trash based only on this argument. There are at least three reasons to reconsider Lanier's view seriously:

- agents have been used for years in aeronautics, and there are lessons learned from this deep experience;
- since our occidental societies are moving from energy-based interaction (sensory-motoric activities) to information-based interaction (cognitive activities), the concept of agent has become extremely important for analyzing this evolution from a socio-cognitive viewpoint;
- the concept of agent needs to be taken in a broader sense than the description provided by Jaron Lanier.

More recently, Ben Shneiderman and Pattie Maes (1997) resumed a debate on direct manipulation versus interface agents that has been ongoing for long time in the intelligent interface community (Chin, 1991). Direct manipulation affords the user control and predictability in their interfaces. Software agents open the way to some kind of delegation. I take the view that software agents make new practices emerge. Software agents are no more than new tools that enable people to perform new tasks. The main flaw in current direct manipulation argumentations is that interaction is implicitly thought with an 'acceptable' level of current practice in mind. Current practice evolves as new tools emerge. When steam engines started to appear, the practice of riding horses or driving carriages needed to evolve towards driving cars. Instead of using current practice based on knowledge of horses' behavior, drivers needed to acquire new practice based on knowledge of cars' behavior. For example, when someone driving a carriage wanted to turn right, he or she needed to pull the rein to the right but according to a very specific knowledge of what the horse could accept, understand and follow. Driving a car, someone who want to turn right simply pull the steering wheel to the right according to a very specific knowledge of what the car could accept, 'understand' and follow. It will not shock anyone to say that today pulling the steering wheel to the right is direct manipulation. However, pulling the rein to the right will not necessarily always cause the expected result for all of us, especially for those who do not know horses very well. This kind of human-horse interaction is obviously agent-based. Conversely, horse riders who discovered car driving at the beginning of the twentieth century did not find this practice very natural compared to riding a horse. In this case, the artificial agent was the car engine that the driver needed to control. Today, new generation commercial aircraft include more artificial agents that constitute a *deeper interface* between the pilots and the mechanical devices of the aircraft. Direct manipulation is commonly thought of as ‘direct’ actions on these physical devices. It is now very well recognized that the pilots who fly new generation commercial aircraft find their job easier than before. Their job is not only a direct manipulation steering task, but a more high-level flight management task. They need to manage a set of intertwined artificial agents that perform some of the jobs that they performed before. The development of artificial agents is a specific automation process. It is much more appropriate to investigate automation issues in terms of acceptability, maturity and emergence of new practices.

I claim that artificial agent design needs more guidance and principles. This article introduces a human-centered approach to agent design that is based on the elicitation and use of cognitive functions that are involved in the performance of tasks intended to be delegated to a computer. Software agents are used to perform a few tasks that are usually performed by people. This delegation process generates the emergence of new supervisory tasks that people need to perform. These new tasks are not necessarily easy to learn, retain and perform efficiently. The first thing to do is to identify these new tasks. They usually lead to new types of human errors and new styles of interaction that also need to be identified.

2. LESSONS LEARNED FROM AERONAUTICS

The agent-orientation of human-machine interaction is not new. Airplane autopilots have been commonly and commercially used since the 1930’s. Such artificial agents perform tasks that human pilots usually perform, e.g., following a flight track or maintaining an altitude. Control theory methods and tools have handled most of such automation. In the beginning, even if computers that handled such tasks were very basic, feedback processes handled by these systems were not basic at all. If there is one thing that people who are involved in the design of agents should be aware of it is certainly the notion of feedback. It seems that computer scientists are currently (re)discovering this notion, or at least they should be! In other words, automation (that is the design of agents) is a complex process that requires particular attention. The idea of having agents designed by lazy programmers is a fallacy, and the danger is precisely there!

Becoming an airline pilot requires a long training time. This is because the airplane can be considered as an agent itself. It took a long time to integrate and validate autopilots in aircraft cockpit. Lots of research has been carried out to better understand how pilots are handling flight qualities both manually and using autopilots. Today, if autopilots are 'trivial' agents on-board, they require specific pilot training. Over the last 20 years, the development of new generation aircraft has enhanced the integration of computers into the cockpit. Software agents, such as flight management systems (FMSs), emerged. Christopher Wickens advocates the fact that this new kind of automation may cause situation awareness problems:

"While the FMS usually carries out its task silently, correctly and efficiently, there are nevertheless a non-trivial number of exceptions. In fact, a frequently quoted paraphrase of pilots' responses to many advance automated systems is: 'what did it do?, why did it do it?, and what will it do next?' (Wiener, 1989; Rudisill, 1994; Dornheim, 1995). These words are verbalizations of 'automation induced surprises', reflecting a lack of situation awareness which has been documented systematically by a series of experimental investigations carried out by Sarter and Woods (Billings, 1991; Sarter & Woods, 1992, 1994), and supported by aircraft incident analyses (Wiener, 1989; Rudisill, 1994), as well as reconstruction of several recent accidents (Dornheim, 1995)" (Wickens, 1996, page 5)

In fact, even if a pilot develops a mental model through training, that enables the anticipation of a large set of both normal and abnormal situations, this mental model may also be degraded by negative effects of system complexity (Wickens, 1996). This kind of degradation is well shown by Sarter and Woods (1994).

Here, I would like to make the point that human kind is distinguished from the other species because it has the capacity and the desire to build tools to extend its capacities. There are various kinds of tools that humans build. Let us call them artifacts. They can be more or less autonomous. They all require both intellectual and physical capacities from their users. Up to this century, most artifacts required more physical capacities than cognitive capacities from users. Today the reverse is true. From the manipulation of 'physical' tools, we have moved towards interaction with 'cognitive' systems. This is the case in aviation as in many advanced industrial sectors and our everyday private life. In addition, artifacts of the past were designed and developed over longer periods of time than now. Our main problem today is speed and thus lack of artifact maturity, i.e., we need to produce artifacts faster and faster. Users also need to adapt to new artifacts faster than before. Fast human adaptation to artifacts that demand even more, often not stabilized, cognitive resources is even more difficult. This is an excellent reason to think more about principles and criteria for a human-centered design of artificial agents. This starts by defining properly what an agent is.

3. WHAT IS AN AGENT?

An agent is an artifact or a person that/who acts. An agent produces actions that produce effects. Agents are taken in the sense of Minsky's terminology (Minsky, 1985). An agent is always associated to a cognitive function. A cognitive function can be interpreted in the mathematical sense or in the teleological sense. The former interpretation leads to the definition of an application transforming an input into an output. The input is usually a required task to be performed. The output is the result of the execution of the task. We usually say that the agent uses a cognitive function that produces an activity or an effective task. The latter interpretation leads to the definition of three attributes of a cognitive function:

- a role, e.g., the role of a postman (i.e., an agent) is to deliver letters;
- a context of validity, e.g., the context of validity of the above role is defined by a time period that is the business hours and a specific working uniform, for example;
- a set of resources, e.g., the resources necessary to execute the function are riding a bicycle, carrying a big bag and performing a delivery procedure, for example. Note that a resource is a cognitive function itself.

Some smart artifacts may not qualify for being artificial intelligence (AI) systems, but they implicitly include the use of appropriate human cognitive function resources that make intelligent the resulting user-artifact system. For example, speed bugs on airplane speed indicators are not intelligent agents in the AI sense, but they are smart artifacts. Speed bugs are set by pilots to anticipate and inform on a decision speed.

Users develop appropriate *cognitive functions* to speed up, and increase both comfort and safety of their job, i.e., the tasks that they usually perform. These cognitive functions can be *soft-coded* or *hard-coded.* When they are soft-coded, they usually appear in the form of procedures or know-how stored in their long-term memory. When they are hard-coded, they usually appear in the form of interface devices or manuals that guide users in their job. In both cases, cognitive functions can be either *implicit* or *explicit.* When they are implicit, they belong to what is usually called *expertise.* When they are explicit, they belong to what is usually called *sharable knowledge.* Sometimes, cognitive functions remain implicit for a long time before becoming explicit and easily sharable. When a cognitive function is persistent, it can be formalized into an artificial agent to improve the performance of the task. This is commonly called *automation.* The development of machine agents increases the levels of automation. Human operators are faced with machine assistants that provide a pseudo-natural interaction.

More generally, an agent can be natural or artificial (artifactual). The former type includes people, therapeutic or atmospheric agents, for example. We try to better understand how they work, and model them in order to better anticipate their actions. The latter type includes automated power plants, sophisticated vehicles, advanced computer networks or software agents, for example. Humans have built them, but it is time to better understand their usability, and model them in order to better control them. A major issue is that artificial agents cannot be studied in isolation from people who are in charge of them.

Automation has been a major concern for a long time. The clock is certainly one of the best example of an old automaton that provides time to people with great precision. People rely on clocks to manage their life. A watch is also a unique artificial agent that provides precise time information to a user. In addition, a clock may be programmed to autonomously alert its user to wake up for example. People trust clocks, but they have also learnt to know when clocks do not work properly. They have learnt to interact with such an agent. No one questions the use of such an agent today. The role of the clock agent is to provide the time to its user. Its context of validity is determined by several parameters such as the working autonomy of the internal mechanism or the lifetime of the battery. Its resources include for instance the use of a battery, the ability of its user to adjust time when necessary or to change the battery. Note that the user is also a resource for the watch artificial agent.

Thinking in terms of agents relies on a distributed-cognition view (Suchman, 1987; Vera & Simon, 1993) rather than a single-agent view (Wickens & Flach, 1988). The distributed cognition paradigm states that knowledge processing is distributed among agents that can be humans or machines (Hutchins, 1995). Sometimes designing an artificial agent that is intended to help a user may not be as appropriate as connecting this user to a real human expert; in this case, the artificial agent is a 'connector' or a 'broker' between people.

4. POSSIBLE AGENT-TO-AGENT INTERACTION

Human-centered design of artificial agents is based on the nature of interaction among both human and artificial agents. The type of interaction depends, in part, of the knowledge each agent has of the others. An agent interacting with another agent, called a partner, can belong to two classes: (class 1) the agent does not know its partner; (class 2) the agent knows its partner. The second class can be decomposed into two sub-classes: (subclass 2a) the agent knows its partner indirectly (using shared data for instance), (subclass 2b) the agent knows its partner explicitly (using interaction primitives clearly understood by the partner). This classification leads to three relations between two agents interacting:

- (A) competition (class 1);
- (B) cooperation by sharing common data (subclass 2a);
- (C) cooperation by direct communication (subclass 2b).

In the *competition* case, the agent does not understand inputs to and outputs from the other agents. This can lead to conflict for available resources. Thus, it is necessary to define a set of synchronization rules for avoiding problems of resource allocation between agents. Typically, these synchronization rules have to be handled by a supervisor, an advisor or a mediator (Figure 1). This agent can be one of the partners or an external agent. It is not necessary to explain its actions and decisions. The other agents rely on it to insure a good interaction.

In the case of *cooperation by sharing common data,* the agent understands inputs to and outputs from the other agents. Both of them use a shared data base (Figure 2). Such a shared data base can be an agent itself if it actively informs the various agents involved in the environment, or requests new information (self updating) from these agents, i.e., it is an explicit mediator. Agents use and update the state of this database. An example would be that each agent note all its actions on a blackboard to which the other agents refer before acting. Agents have to cooperate to use and manage the shared database. This paradigm leads to a data-oriented system. Such a system has to control the consistency of the shared data. Cooperative relations between agents do not exclude competitive relations, i.e., resources for which the corresponding agents may be competing generally support shared data. In this case, synchronization rules have to deal with resource allocation conflicts and corresponding data consistency checking.

In the previous cases, the interaction is indirect. In the case of *cooperating by direct communication,* agents interact directly with the others (Figure 3). They share a common goal and a common language expressed by messages, e.g., experts in the same domain cooperating to solve a problem. We say that they share a common ontology, i.e., common domain and task models. When this knowledge sharing is not clearly established, cooperation by direct communication is hardly possible: agents do not understand each other. An artificial agent that satisfy this type of relation must then include a user model (Mathé & Chen, 1996).

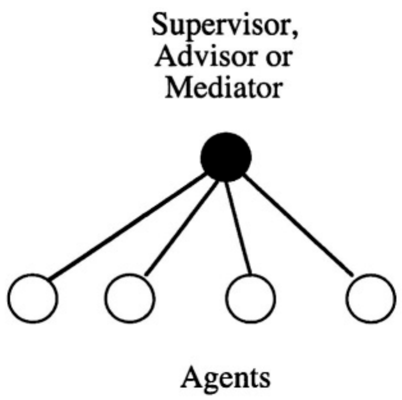

Figure 1. Competition: agents need to have a supervisor, an advisor or a mediator to help manage their interactions.

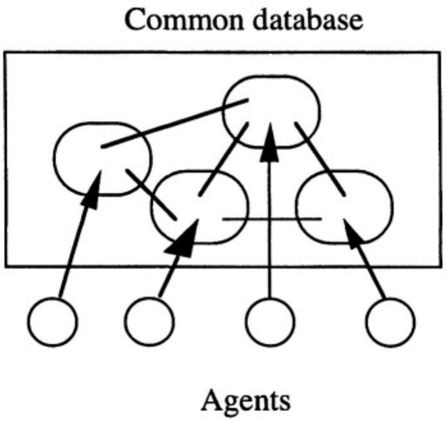

Figure 2. Cooperation by sharing common data: agents manage to communicate through a common database that is an interface between the agents.

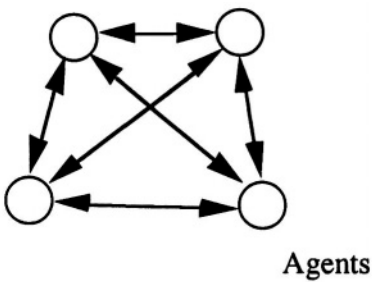

Figure 3. Cooperation by direct communication: agents interact directly with each other.

5. AN ECOLOGICAL APPROACH: LOOKING FOR MATURITY

In this section, I explain why my research agenda is not in the current main stream of the software agent community. I am not interested in the way an agent is developed from a software engineering perspective. I am interested in the way a software agent is being used, modifies current work practice, influences the environment and work results (i.e., products), and modifies ways evaluation/certification is currently performed for non-agent systems. I also try to start a theoretical discussion on what artificial agents really are or will be. I realize that I am in the same position as Jules Verne who described the way people might use a submarine one century ago long before submarines were built and operated as they are now. In other words, I am interested in exploring where we are going by developing and using software agent technology. This article takes some of the factors that Norman provided on how people might interact with agents (Norman, 1994).

6.1Hiding unnecessary complexity while promoting necessary operations

Prior to the integration of flight management systems (FMSs) onboard aircraft, pilots planned their flights using paper and pencil technology. An FMS is a real-time database management system where flight routes are stored. It enables the pilot to program or recall a flight route and adapt it to the current flight conditions. This machine-centered flight management is programmed to define a vertical profile and a speed profile, taking into account air traffic control requirements and performance criteria. Once a flight route is programmed into the system, the FMS drives the airplane by providing setpoints to the autopilot. The FMS computes the aircraft position continually, using stored aircraft performance data and navigation data (FCOM-A320, 1997). The same kind of example was studied by Irving et al. using the GOMS approach (Irving et al., 1994), and experimentally by Sarter and Woods to study pilots' mental load model and awareness of the FMS (Sarter & Woods, 1994). "While most pilots were effective in setting up and using the FMS for normal operations, a substantial number revealed inadequate situation awareness under conditions when the system would be unexpectedly configured in an unusual, but not impossible, state. These configurations might result from an erroneous pilot input, from the need to respond to unexpected external events (e.g., a missed approach), or from a possible failure of some aspect of the automation. Under these circumstance, a substantial number of pilots simply failed to understand what the FMS was doing and why; they were surprised by its behavior in a way that would make questionable their ability to respond appropriately." (Wickens, 1996, page 5). Designers have created a large number of options to control the FMS complexity. For example, there are at least five different modes to change altitude. A software agent that would provide the right one at the right time and in the right understandable format to the pilot would be very valuable. This requires an event-driven approach to design, i.e., categories of situations where pilots would need to use an appropriate mode to change altitude, for example, should be clearly elicited and rationalized.

One of the main reasons why the event-driven approach is not frequently taken is because it is very expensive in time and money. Today engineering rules business. Engineers have a goal-driven approach to design, and they unfortunately often end up with externally complex user interfaces. Technology is evolving very fast due to smart engineers who continually improve artifacts without crossing their views with other professionals such as marketing experts, usability specialists and scientists. “Development is a series of tradeoffs, often with incompatible constraints.” (Norman, 1998). This is even more true for the development of artificial agents and automation in general. If artificial agents are developed to decrease user workload or increase safety, they also tend to decrease vigilance and increase complacency (Billings, 1991). This is why cognitive function allocation is fundamental in the design process of an artificial agent: What new supervisory functions will it require from users? What situation awareness functions will it make emerge in various situations? What will be the most appropriate interaction functions that will need to be implemented in its user interface? Since such a cognitive function analysis needs to be carried out very early during the design process, the development process (and the company) should be re-organized, as Don Norman already suggested (1998).

6.2 Affordance: The ultimate maturity of an artifact

I don’t want to question the main attributes of software agents provided by Pattie Maes such as personalization, proactivity, continuous activity, and adaptivity (Shneiderman & Maes, 1997). They are fine, and I am very comfortable with them as they match good technology-centered automation. However, they are not sufficient. Maturity is a key issue for automation, and high-technology in general. “... look around us at those high-technology products... ask why so many telephone help lines are required, why so many lengthy, expensive phone calls to use the product... go to a bookstore and look at how many bookshelves are filled with books trying to explain how to work the devices. We don’t see shelves of books on how to use television sets, telephones, refrigerators or washing machines. Why should we for computer-based applications” (Norman, 1998). This is where the concept of affordances needs to be considered seriously. An artificial agent needs to be affordable to its user in any workable situation.

The term "affordances" was coined by James Gibson to describe the reciprocal relationship between an animal and its environment, and it subsequently became the central concept of his view of psychology, the ecological approach (Gibson, 1979). In this article, affordances are resources or support that an artificial agent offers to its user; the user in turn must process the capabilities to perceive it and use it. How do we create affordances for an artificial agent? Don't expect a simple and clear procedure for doing this. It will be an iterative cycle process of design, engineering, evaluation and analysis. However, better understanding the procedure-interface duality is key towards the incremental discovery of agent affordances. Agent affordances deal with intersubjectivity, i.e., the process in which mental activity is transferred between agents. A mental activity could be situation awareness, intentions, emotions or knowledge processing for example.

People interacting with artificial agents usually follow operational procedures in either normal or abnormal situations. Operational procedures can be learned in advance and memorized, or read during performance. Think about the operational procedure that you need to follow when you program your washing machine or your VCR. Operational procedures are supposed to help operators during the execution of prescribed tasks by enhancing an appropriate level of situation awareness and control. It is usually assumed that people tend to forget to do things or how to do things in many situations. Procedures are designed as memory aids. In abnormal situations for example, pilots need to be guided under time-pressure, high workload and critical situations that involve safety issues. Procedures are often available in the form of checklists that are intended to be used during the execution of the task (it is shallow knowledge that serves as a guideline to insure an acceptable performance), and operations rationale that needs to be learned off-line from the execution of the task (this is deep knowledge that would induce too high a workload if it was interpreted on-line.) The main problem with this approach is that people may even forget to use procedures! Or they anticipate things before the execution of a procedure. People tend to prefer to use their minds to recognize a situation instead of immediately jumping on their checklist books as they are usually required to do in aviation, for instance (Carroll et al., 1994). In other words, people are not necessarily systematic procedure followers (De Brito, Pinet & Boy, 1998). They want to be in control (Billings, 1991). Ultimately, if the user interface includes the right situation patterns that afford the recognition of and response to the right problems at the right time, then formal procedures are no longer necessary. In this case, people interact with the system in a symbiotic way. The system is affordable. The better the interface is, the less procedures are needed. Conversely, the more obscure the interface is, the more procedures are needed to insure a reasonable level of performance. This is the procedure-interface duality issue.

6.3 Discovering affordances using active design documents

By designing concurrently an artificial agent and its operational procedures from the early stages of the design process, affordances are more likely to emerge incrementally.

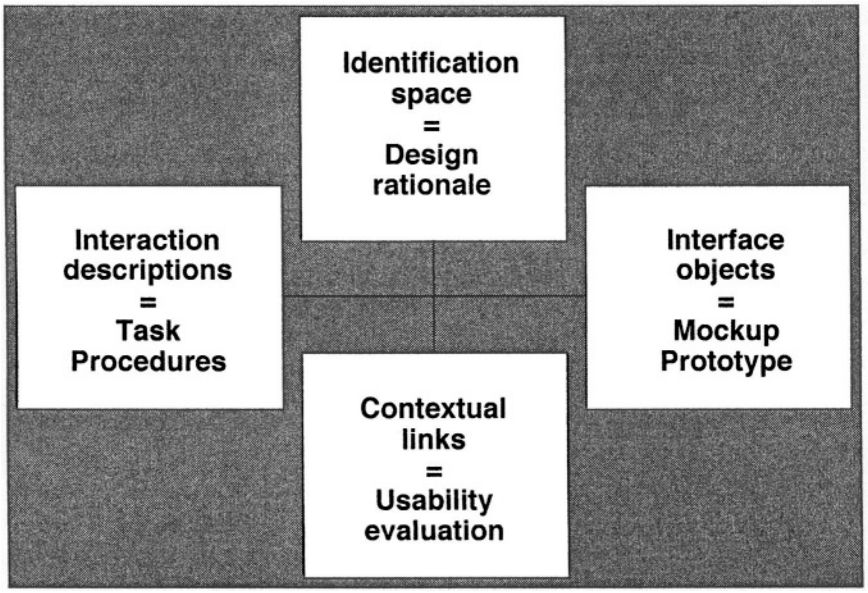
Identification space = Design rationale
Interaction descriptions = Task Procedures
Interface objects = Mockup Prototype
Contextual links = Usability evaluation

Figure 4. A generic active design document.

This is a reason why we have already proposed the active design document approach to support this process (Boy, 1998). An active design document includes four aspects (Figure 4):

- *interaction descriptions*–the symbolic aspect, which conveys ideas and information, e.g., the description of a procedure to follow; this aspect of an active design document is related to the task involved in the use of the artifact; it defines the *task space;*
- *interface objects* connected to interaction descriptions–the emotive aspect, which expresses, evokes, and elicits feelings and *attitudes,* e.g., a mockup of the interface being designed; this aspect is related to the interface of the artifact that provides interactive capabilities; it defines the *activity space;* note that interface objects are characterized by specific

cognitive functions (to be elicited incrementally by a series of usability evaluations) provided to the user to improve interaction;
- *contextual links* between the interaction descriptions and the interface objects, e.g., annotations or comments contextually generated during tests; this aspect is related to the user and the environment in which the artifact is used; it defines the *cognitive function space.*
- *an identification space*; in addition to its three definitional entities, i.e., interaction descriptions, interface objects, and contextual links, each active design document is identified by an identification space that includes a name, a list of keywords, a date of creation, a period of usability tests, a design rationale field and a set of direct hypertext links to others active design documents.

The development of active design documents is incremental and fosters participatory design. They enable the design team to minimize the required complex procedures to be learned, and maximize affordances of the artificial agent being designed. A traceability mechanism enables anyone to figure out at any time why specific affordances have emerged (Boy, 1999).

6.4 Human-centered design of artificial agents

Understanding the needs of potential users of artificial agents does not consist in asking them what they want. They usually don't know this, or even worse they think they know! Facing them with a prototype and asking them what they think of it is much better. This is what usability testing is about. This is why incremental development of active design documents is likely to generate good affordances. Users enter into the design process to provide their views on perceivable interface objects that enable them to generate an activity using the agent, and on attached interaction descriptions that enable them to guide this activity. Contextual links are filled in after each evaluation, and used to redesign the agent. Each time a design is produced, the design rationale is stored in the identification space.

Designing for simplicity is key. Artificial agents need to be simple, easily understandable, and fun to use. This does not mean that people will not have to learn new values and skills by using them. Using artificial agents looks like getting a promotion at work. You now manage a group of agents that work for you. New management skills are thus necessary. This changes work practice that needs to be addressed during the design process. The job will not be the same as before. In particular, creating artificial agents involves new cooperation and coordination processes that were not relevant before. Questions are: How different will the job be? How difficult will it be to learn it? Will it require 'new' people?

6.5 Adapting Henderson's design cycle to agents

Austin Henderson brought a very interesting distinction of design from science and engineering (Ehrlich, 1998). Science brings rationalization of current practice (Boy, 1998, page 190). Science tries to understand where we are now. Let us acknowledge that agent science is very preliminary. Design is where we would like to be. It is an exercise of imagination. For the last few decades designers have been very prolific in imagining and inventing new intelligent devices that lead to agents. Designers ask specific questions such as: "What direction can we go in? Where might that take us? What would the implications be? " (Ehrlich, 1998, page 37). Engineering addresses how do we get from here to there taking into account the available resources. Once engineers have developed new artifacts, science takes the lead again to figure out where we are according to the emergence of new practices introduced by these new artifacts (Figure 5).

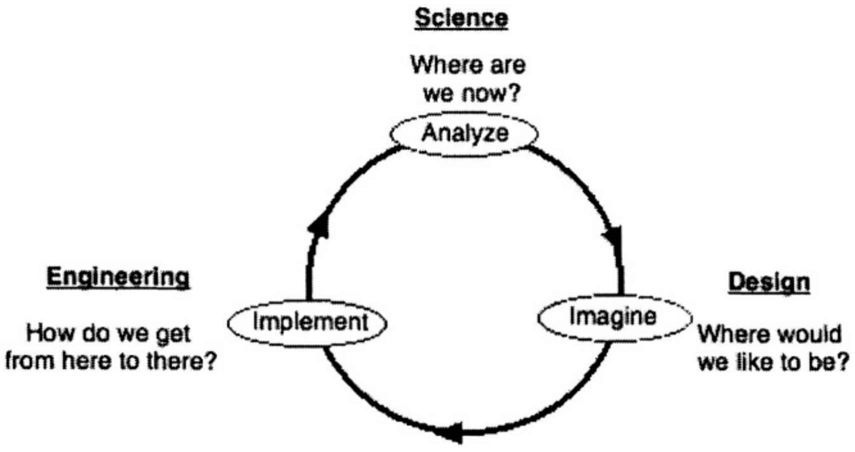

Figure 5. Henderson's cycle.

Most current software agent contributions address the engineering perspective. Since we are still very poor in agent science, it is very difficult to address properly the design perspective from a humanistic viewpoint. Although it is true that new practices that emerge from the use of artificial agents constitute very important data that science needs to analyze and rationalize. Experience feedback on the use of agents is still very preliminary. A good way to address the design perspective today is to develop participatory design (Muller, 1991) involving end-users in the design process. In addition, there will be no human-centered design of artificial agents without an appropriate set of usability principles. Several traditional human factors principles and approaches have become obsolete because the paradigm of a single agent, as an information processor, is no longer appropriate in a multi-agent world. Multi-agent models are better suited to capture the essence of today's information-intensive interaction with artificial agents. Many authors working in the domain of highly automated systems described agent-to-agent communication (Billings, 1991; Hutchins, 1995). A human agent interacting with an artificial agent must be aware of:

- what the other agent has done (history awareness);
- what the other agent is doing now and for how long (action awareness);
- why the other agent is doing what it does (action rationale awareness);
- what the other agent is going to do next and when (intention awareness).

These four situation awareness issues correspond to the most frequently asked questions in advanced cockpits (Wiener, 1995). In order to describe human-computer interaction several attributes are already widely used such as the basic usability attributes proposed by Nielsen (1993). From our experience in aeronautics, the following attributes were found important in multi-agent human-machine communication (science contribution in Henderson's sense):

- prediction, i.e., ability to anticipate consequences of actions on highly automated systems;
- feedback on activities and intentions;
- autonomy, i.e., amount of autonomous performance;
- elegance, i.e., ability not to add additional burden to human operators in critical contexts;
- trust, i.e., ability to maintain trust in its activities;
- intuitiveness, i.e., expertise-intensive versus common-sense interaction;
- programmability, i.e., ability to program and re-program highly automated systems.

6. AN EXAMPLE OF COGNITIVE FUNCTION ANALYSIS

To effectively design and use artificial agents, researchers, designers and engineers must grapple with a number of difficult questions such as: What kinds of tasks are best performed by humans or computers? What are the practical limits of system autonomy? and, Who should be in control?

The development of an artificial agent is based on an incremental process of design/evaluation. In the cognitive function analysis methodology, this process uses the Artifact-User-Task-Organizational Environment (AUTO) pyramid (Boy, 1998). In this approach to designing artificial agents, the analysis and design of cognitive systems is viewed in the light of the linked human-centered-design dimensions of *artifact* (artificial agent), *user, task* and *organizational environment.* The dimensions of user, task and artifact are the factors that are normally taken into account in system design. The dimension of organizational environment enriches the framework, encompassing as it does, roles, social issues, and resources.

Let us use an example from our everyday life, as the domain complexity of the aircraft flight deck to which the approach has been previously applied can obscure the principles of the cognitive function analysis. My point is to demonstrate, with selectively intuitive examples, that cognitive function analysis can systematically generate design alternatives based on allocation of cognitive functions (CFs). This example is given as clocks or watches are frequently used, often for very simple tasks such as setting time.

Informal enquiry revealed that many users have watches with knob-hand-display arrangements similar to that presented in Figure 6, often report confusing the selection of which knob turns on which hand or display *every time* they use their watches. Unsurprisingly users report this to be frustrating. We use this opportunity to show how different allocations of cognitive function affect design and use. Setting the minutes and hours requires the user to select the hand by pulling the knob entirely and turning it until the required time is set. Setting the week day and month day requires the user to select the right display by pulling the knob entirely, pushing it a little, and turning it right for the week days and left for the month days until the required time is set. People have difficulties finding the intermediary position, and the right direction to turn (right or left) to set weekdays and month days.

6.1 Design case 1: Allocation of cognitive functions to User

The watch in Figure 6 has a straightforward role and its affordances are clear. Even without a formal task analysis, it is reasonably clear that accomplishing the goal of setting the time, and executing the cognitive function of *setting the time to the required time* (on this watch), the user's tasks are to: *choose to change minute, hour, week day or month day, select the hands or the right display, turn the knob until the right time is set.* The resulting operation seems extremely simple. However, the user problems reported with this design case indicate that the watch does not afford the ready completion of the cognitive function *setting the time to the required time,* as performance breaks down at the task of *select the hands or the right display.*

The performance of these tasks is linked to the achievement of the goal of *setting the time to the required time* largely through the design of the watch. We can change the design and change the tasks, which is a good idea as we have task-related problems, and the goal will be met as long as the AUTO resources (artifact, user, task and organizational environment) still somehow collectively perform the required cognitive function. The watch design case in Figure 6 does not afford the task because the *knob function selection* cognitive function required of the user to find the right knob position and direction of turn is not afforded by the layout of the multi-function knob. A lack of functional retention detracts from the affordances of this watch. In this design case the user must work with the artifact (through experimentation) to accomplish the task and thus the goal. In cognitive function terms, there is a disjunction here between the prescribed task (what the designer had in mind) and the activity that is actually performed by the user. For example, some users who need to select the day display don't pull entirely the knob before finding the intermediary position, they pull the knob to the intermediary position directly and they are frustrated to observe that turning right or left does not produce any week day or month day modification. This disjunction is revealed as the performance of an added task (that of 'experimentation') to achieve the cognitive function of *knob function selection,* so that the prescribed task of performing the right sequence of *selecting* and *turning* the correct knob position can be achieved. The observed divergence between prescribed task and activity, combined with user feedback tells us that the allocation of cognitive functions amongst the AUTO resources needs redesign. The repeated 'experimental' nature of the activity informs us that it is the artifact that will benefit from redesign.

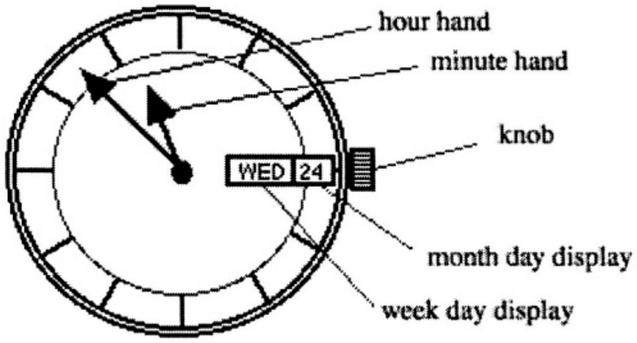

Figure 6. A 'classical' single knob watch.

This cognitive function allocation solution induces a competition process between the user and the artifact. The cognitive functions that are implemented in the watch are engineering-based. For example, the multi-function knob is a very clever piece of engineering since with a single device one can set four time parameters. The main problem is that the end-user needs to be as clever as the engineer who designed the device in order to use it successfully, or use an operation manual that will help supervise the user-watch interaction.

6.2 Design case 2: Allocation of cognitive functions to User and Artifact

In the second design case (Figure 7a), there is a knob for each function (minutes, hours, week days and month days). This alternative design removes part of the selection confusion. The user needs to know that the upper-right knob is the hour-setting knob, and so on as shown on Figure 7a. There is a pattern-matching problem. This design can be improved if the knobs are explicitly associated with the data to be set.

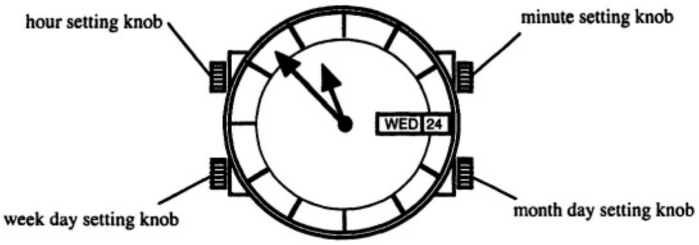

Figure 7a. A multi-knob setting watch.

Figure 7b presents a digital watch interface that removes the requirement for identifying *which knob operates which hand or display* from the user—and with it the cognitive function of pattern matching. The knob-display relationship has become an explicit feature of the watch that exploits existing user attributes and affords selection of the correct knob. The user's task is now simply to select the knob that is next to the time data to be set, and to turn this knob.

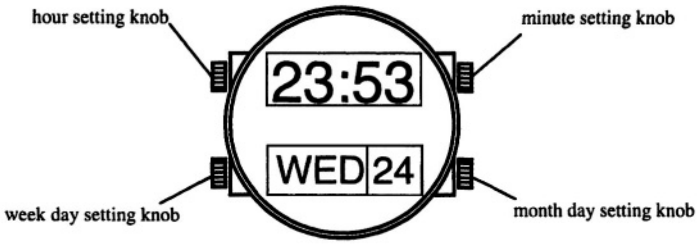

Figure 7b. Associative setting watch.

This cognitive function allocation solution induces cooperation by sharing common data between the user and the artifact. Each time-setting device is associated to a single function that the end-user understands immediately such as in the design case shown in Figure 7b. The small physical distance between the time-setting knob and the corresponding data display makes this possible. The end-user does not need an operational manual.

6.3 Design case 3: Allocation of cognitive functions to Artifact

In the third example (Figure 8), new technology is used to design the watch, which has the characteristic of setting time automatically in response to a voice command such as 'set the time to 23:53', 'set the week day to Wednesday', or 'set the month day to 24'. We have transferred The *select the hands or the right display, turn the knob until the right time is set* part of the cognitive function of *setting the time to the required time* is transferred to the watch. The user's task has now become that of simply pushing the voice button and talking to the watch. But, because the whole cognitive function is not transferred to the watch, the user must still perform the cognitive function of *to the required time.* This requirement results in the task of 'looking at the data being set'. Designing an artificial agent that recognizes the speech of the user is not trivial since it needs to take into account possible human errors such as inconsistencies.

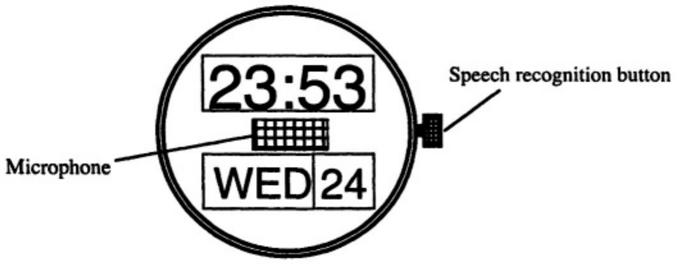

Figure 8. Automated-setting watch

This cognitive function allocation solution induces cooperation by direct communication between the user and the artifact. The watch speech recognition and natural language understanding artificial agent needs to interpret what the user is saying. It needs to filter noisy data, remove inconsistencies, ask follow-up questions in the case of misunderstanding (i.e., no pattern-matching with available patterns). This means that the corresponding artificial agent should include a user model.

We could also transfer this remaining cognitive function to the artifact by designing a radio receptor or a datalink device, which the user could trigger to get the time from a national time service, transferring authority for *the* *right time* to the artifact. The users task would now simply be to push the time setting button. Thus almost the entire cognitive function for achieving the goal has been transformed and transferred to the watch. Still the user may verify that the data transfer was properly done, and eventually push on the button again.

Several issues emerge from this design case, firstly that a set of inappropriate affordances has been established. This design affords being used inappropriately, for example performed in a geographical zone that is not equipped with this service.

6.4 Design case 4: Allocation of cognitive functions to Organizational Environment

Finally we consider allocating the entire cognitive function for setting the time to the AUTO environmental resource (See Figure 9). Thus, instead of providing a watch-setting device, a direct datalink connection is available on the watch, i.e., the user does not have anything to do, the watch is automatically set by taking data from the above-mentioned national service when necessary and possible.

This design case is the ultimate automation solution. User acceptance of this solution depends on the reliability of the datalink system. The user will learn what the flaws of this design solution are and adapt to them. For instance, when he or she goes on vacations in a region where the datalink connection does not work, either he or she will not care about time setting, or he or she will use another more traditional watch.

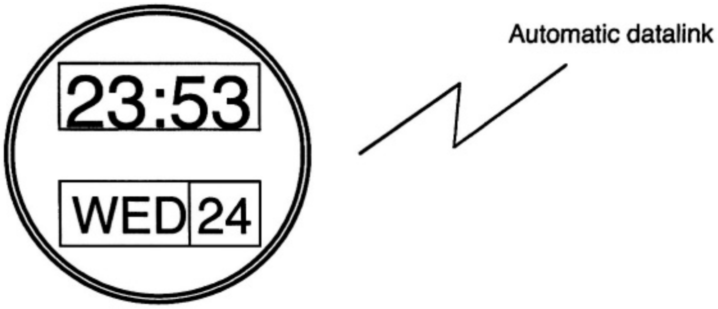

Figure 9. No time setting device (automatic datalink).

6.5 Analysis and evaluation

The nature of the interactions among the four types of design is quite different. Consequently the artifact and user cognitive functions of the systems are also different, yet they all enable the system to meet the goal. In the first design case, the user is a problem solver, in the second he or she needs only a little artifact knowledge, in the third he or she manages an artificial agent, and in the forth he or she delegates.

The AUTO pyramid helps the analyst decide which resources are relevant and important, and assists the designer in establishing appropriate design options. However, to obtain some objectivity, consistency and traceability it is important to evaluate the designs using a significant task and an appropriate set of evaluative criteria. The evaluation is performed on the *time-setting* task. Table 1 provides an evaluation of the watch design cases over attributes that were found important in multi-agent human-machine communication (Boy, 1998).

***Table 1.* CFA evaluation criteria ratings over watch design cases**

Criteria for the Setting-time task	Watch Design Cases				
	1	2a	2b	3	4
Prediction	High	High	High	High	High
Feedback	Low	High	High	High	N/A
Autonomy	Low	Low	Low	High	High
Elegance	Low	Mediu	High	Mediu	High
Trust	High	High	High	Mediu	High
Intuitiveness	Low	Mediu	High	High	High
Programmabilit	Mediu	High	High	Mediu	N/A

Design case 1, i.e., a classical single knob watch, enables the interpretation of the first attribute (prediction) in the sense of simplicity and habit of use. In this sense, the time-setting task is very predictable,. In addition, even if several errors are possible, they are predictable. Feedback is low when the user tries to set a time and nothing happens. There is no indication of a bad mode selection for example. In addition, there is no indication on how to recover from dead-ends. Autonomy is low because the user needs to manually perform the time-setting task. Elegance is also low since the fact that human errors are very likely in any situation, they will not ease the overall process in critical contexts. Trust is high when the time-setting mechanism is working properly. The use of the single knob device is not intuitive, even if it is based on a simple design. Once the user has selected the right mode (ability to understand what to do), programming is easy (ability to perform the task efficiently).

Design case 2a, i.e., a multi-knob setting watch, is not significantly different from design case 1 as far as prediction, autonomy and trust are concerned. However, feedback is high since when the user uses any knob the result is always observable on the watch. Elegance is medium and better than in design case 1 because in critical contexts any human error can be detected rapidly for instance. Analogously, intuitiveness is medium because associations can be made between a button and a hand. Programmability is high because oce the right button is selected, it is easy to set time.

Design case 2b, i.e., associative setting watch, is a major improvement to the design case 2a since the watch is more affordable in terms of elegance and intuitiveness.

Design case 3, i.e., automated-setting watch, keeps high prediction, feedback and high intuitiveness. Its major improvement on the previous design alternatives is its high autonomy. However, it has some drawbacks. In particular, elegance is medium because in critical contexts voice could be different from the regular voice used in normal operations for instance. The complexity of the interpretation performed by the voice recognition system might induce errors that may lead to trust problems in the long term. Programmability is medium since the calibration of the voice recognition system might not work in all situations.

Design case 4, i.e., no time setting device (automatic datalink), does not require any action from the user. All evaluation criteria are rated high if the datalink system is very reliable, except the feedback and programmability attributes that are not applicable (N/A) in this case.

7. INTERPRETATION VERSUS AMPLIFICATION

A modern artifact such as the watch shown in Figure 8 can be defined as a cognitive system, i.e., it includes a software agent that constitutes a *deeper interface* between a mechanical artifact and a user. A software agent is a new tool mediating user-artifact interaction. The physical interface is only the surface of this deeper interface. A current research topic is to better understand what operators need to know of this deeper interface. Should they only know the behavior of the physical interface? Should they understand most of the internal mechanisms of the deeper interface? How should the deeper interface represent and transfer the behavior and mechanisms of the (mechanical) artifact?

From a philosophical viewpoint, the issue of user-(mechanical)artifact systems can be seen as whether the coupling is between the (mechanical) artifact and the software agent (Figure 10a) or between the software agent and the user (Figure 10b).

The distinction between *interpretation* and *amplification* is important because it entails two completely different views of the role of the user in user-artifact systems, hence also on the design principles that are used to develop new systems. In the interpretation approach, the software agent can be seen as a set of illusions re-creating relevant artifact functionalities; the user sees a single entity composed of the artifact augmented by the software agent. In the amplification approach, the software agent is seen as a tool or an assistant; the couple user/software-agent works as a team to control the artifact.

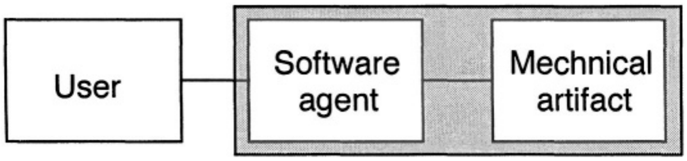

Figure 10a. Interpretation: Software agent replaces user functions.

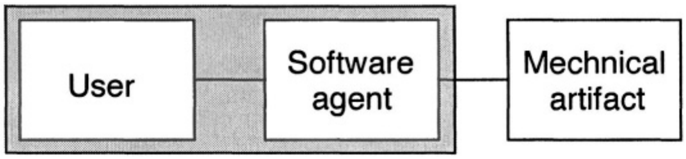

Figure 10b. Amplification: Software agent enhances user capabilities.

Back to the direct-manipulation versus interface agents debate, interpretation induces direct manipulation, and amplification induces delegation. Let us take two examples to illustrate these two approaches:

- The file deletion function, for example, is interpreted by the manipulation of a trash icon of a desktop interface. The trash icon is the visible part of a very simple software agent that incorporates the cognitive function of deleting a file when the user drags a file icon on the trash icon. Other cognitive functions, such as highlighting the trash icon when the file is ready to be included in the trash, facilitate user manipulation by increasing accuracy and understanding. This type of reactive agent removes the burden of remembering the syntax of the delete function from the user for example. The resulting interpretation mechanism improves the affordances of the delete function for the user, and the transmission of a manipulation action of the user on the interface to the machine in the form of a machine-understandable delete function.
- Another type of artificial agent is an on-line spelling checker that informs the user of typos directly when he or she generates a text. In this case, the user delegates spelling checking to such an artificial agent. In a sense, it amplifies the spelling checking user's capability. The coordination of such an artificial agent with the user is crucial. It may result that this kind of artificial agent might be disturbing for the user if after most words it proposes a correction. In this case, the artificial agent should be taking into account the context in which a word is generated. This is very

difficult especially if a safe and mature mechanism is targeted. This is why a human-centered approach to the design of such an artificial agent is required. A very simple cognitive function analysis shows that, user's cognitive functions such as interruption handling during idea generation and development (when the user is typing) is extremely disturbing for the user. Too much interruption handling may cause that the user will turn the spelling checker off, and will not use it on-line. Appropriate responses to this issue would be telling the user that the new augmented system provides a new way of generating text, and requiring that he or she follows a substantial training if necessary. The spelling checker artificial agent needs to be taken into account as an amplification mechanism that needs to be learnt and controlled. In particular, the user should be able to use it for a set of common words that he or she will use often; this needs preparation, and then involves a new way of interacting with the resulting text processor. The artificial agent must not alert the user at all times when the user makes a typo, but wait that a sentence or a whole paragraph is typed, for example.

8. CONCLUSION AND PERSPECTIVES

The watch example showed a typical evolution of current artifacts toward more integration in a technology-centered world. Watches will be automatically set using a global positioning system (GPS) via satellite. The adaptation of artifacts will be done through the use of artificial agents.

We live in an information-intensive world where one crucial issue is not information availability but access to the right information at the right time, in the right format. We usually have much more information available than we need and are able to process. Artificial agents also emerge as a necessity to handle this difficult problem of contextual access to information at the same time of a technological glue between modern artifacts and human beings. The concept of artificial agent itself needs to be thought in a broader sense than the usual software agent sense that the AI community currently proposes.

In a general sense, design is guided by the incremental definition and satisfaction of a set of constraints. An important issue is to make the constraints explicit enough to guide the decisions during the design process. These constraints may be budget-based, technology-based or human-factors-based. Budget-based constraints forces faster design and development processes. It results that current technology does not have enough time to become mature before it is replaced by new technology. In addition, I claim that human operators will experience several changes in their professional life. This is not only due to technology changes but also to job changes. It results that training is becoming a crucial issue. In particular, training is no more only a matter of an initial learning phase, but is becoming a life-time continuous education process that is based on performance support through the use of artificial agents. Even if initial training (including theoretical courses) enables the acquisition of conceptual frameworks, artificial agents could provide hands-on training with the possibility of zooming into deeper knowledge.

Artificial agents for training are not the only types of agents. As a matter of fact, a typology of artificial agents based on their use would be extremely useful. I propose an example of typology that will serve both as a starting reference and an illustration of various potential properties of agents:

- agents that enhance information access (database managers);
- agents that deal with situation awareness (secretaries, error-tolerent/error-resistent assistants or rescuers);
- agents that help users to learn (intelligent tutors);
- agents that enhance cooperative work (connectors or brokers);
- agents that perform tasks that people would not be able to perform without them (cognitive prostheses, workload relief systems);
- agents that learn from interaction experience (learning mechanisms);
- agents that require user's expertise or pure common sense for efficient and safe interaction (specialized versus public agents).

Human-factors constraints need to be more taken into account. In particular, what matters is the type of interaction that agents use to communicate among each other:

- The user does not understand what the artificial agent is doing, and it is very likely that both agent end up with competing. This is why rigid procedures are needed to coordinate agent interaction.
- The user interacts with the artificial agent through a common set of perceivable artifacts that each of them understands. A common vocabulary is used.
- Both the user and the artificial agent are able to understand the rationale of the utterances of the other. A common ontology needs to be shared. In this case, an ontology is an organized framework of cognitive artifacts that may take the form of abstract concepts or concrete devices.

These three types of interaction may be possible in various contexts using the same artificial agent. Context is truly the key issue. Context may be related to the type of user, environment, organization, task and artifact. This is why I have developed the AUTO pyramid that supports human-centered design by providing an integrated framework of these key contextual attributes. The design of an artificial agent should be based on the elicitation of the cognitive functions involved in the user-artifact interaction to execute a task in a given organizational environment. With respect to the AUTO pyramid, cognitive function resources can be user-based (e.g., physiological capabilities and limitations, knowledge and skills), task-based (e.g., checklists or procedures), artifact-based (e.g., artifact level of affordances) or organizational-environment-based (e.g., environmental disturbances, delegation to other agents). Human-centered design of artificial agents is a crucial issue that deserves more investigation and practice.

9. ACKNOWLEDGMENTS

Hubert L'Ebraly, Thierry Broignez, Meriem Chater, Mark Hicks, Christophe Solans and Krishnakumar greatly contributed to the current state of the CFA methodology at EURISCO, Aerospatiale and British Aerospace. Thank you all.

10. REFERENCES

Billings, C.E., 1991, Human-centered aircraft automation philosophy. NASA TM 103885, NASA Ames Research Center, Moffett Field, CA, USA.

Boy, G.A., 1998a, *Cognitive function analysis.* Ablex, distributed by Greenwood Publishing Group, Westport, CT.

Boy, G.A., 1998b, Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems in Proceedings of CHI'98, ACM Press, 265-272.

Chin, D.N., 1991, Intelligent interfaces as agents. In *Intelligent User Interfaces,* J.W. Sullivan and S.W. Tyler (Eds.). ACM Press, New York, U.S.A, pp. 177-206.

De Brito, G., Pinet, J. & Boy, G.A., 1998, *About the use of written procedures in glass cockpits: Abnormal and emergency situations.* EURISCO Technical Report No. T-98-049, Toulouse, France.

Dornheim, M.A., 1995, Dramatic incidents highlight mode problems in cockpits. *Aviation Week and Space Technology,* Jan. 30, pp. 57-59.

Ehrlich, K., 1998, A Conversation with Austin Henderson. Interview. *Interactions. New visions of human-computer interaction.* November/December.

FCOM-A320, 1997, *Flight Crew Operation Manual A320.* Airbus Industrie, Toulouse-Blagnac, France.

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

Hutchins, E., 1995, How a cockpit remembers its speeds. *Cognitive Science,* 19, pp. 265-288.

Irving, S., Polson, P. & Irving, J.E., 1994, A GOMS analysis of the advanced automated cockpit. *Human Factors in Computing Systems. CHI'94 Conference Proceedings.* ACM Press, 344-350.

Lanier, J,, 1995, Agents of Alienation, *interactions.* July, pp. 66-72.

Mathé, N. & Chen, J.R., 1996, User-centered indexing for adaptive information access. *User Modeling and User-Adapted Interaction.* 6(2-3), pp. 225-261.

Minsky, M., 1985, *The Society of Mind.* Touchstone Books. Simon & Schuster, New York.

Muller, M., 1991, Participatory design in Britain and North America: Responding to the «Scandinavian Challenge». In *Reading Through Technology, CHI'91 Conference Proceedings.* S.P. Robertson, G.M. Ohlson and J.S. Ohlson Eds. ACM, pp. 389-392.

Norman, D.A., 1994, How might people interact with agents. *Communications of the ACM,* July, Vol.37, No. 7, pp. 68-71.

Norman, D.A., 1998, *The invisible computer.* MIT Press.

Rudisill, M., 1994, Flight crew experience with automation technologies on commercial transport flight decks. In M. Mouloua and R. Parasuraman, Eds.), *Human Performance in Automated Systems: Current Research and Trends,* Hills dale, NJ, Lawrence Erlbaum Associates, pp. 203-211.

Sarter, N.B. & Woods, D.D., 1994, Pilot interaction with cockpit automation II: An experimental study of pilots' model and awareness of the flight management system. *International Journal of Aviation Psychology,* 4, 1, pp. 1-28.

Shneiderman, B. & Maes, P., 1997, Direct manipulation versus interface agents. *interactions.* November-December issue, pp. 42-61.

Suchman, L., 1987, *Plans and situated actions: The problem of human-machine communications.* New York: Cambridge University Press.

Vera, A. & Simon, H., 1993, Situated actions: A symbolic interpretation. *Cognitive Science,* 17, pp. 7-48.

Wickens, C.D., 1996, Situation awareness: Impact of automation and display technology. *NATO AGARD Aerospace Medical Panel Symposium on Situation Awareness: Limitations and Enhancement in the Aviation Environment* (Keynote Address). AGARD Conference Proceedings 575.

Wickens, C.D. & Flach, J.M., 1988, Information processing. In E.L. Wiener & D.C. Nagel (Eds.), *Human Factors in Aviation.* San Diego, CA: Academic Press, pp. 111-155.

Wiener, E., 1989, *Human factors of advanced technology 'glas cockpit' transport aircraft.* Technical Report 117528. NASA Ames Research Center, Moffett Field, CA.