Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems

Guy A. Boy
European Institute of Cognitive Sciences and Engineering (EURISCO)
4, Avenue Edouard Belin, 31400 Toulouse, France
+33.5.62.17.38.38
boy@onecert.fr

ABSTRACT

The Cognitive Function Analysis is a methodology supported by a mediating tool for the human-centered automation of safety-critical systems [4]. It is based on a socio-cognitive model linking the artifact being designed, the user's activity, the task to be performed, and the organizational environment. Cognitive functions can be allocated to humans or machines. They are characterized by their role, context definition and associated resources. The methodology is supported by active design documents as mediating representations of the artifact, the interaction description and cognitive function descriptors being designed, redesigned and used as usability criteria to evaluate the distribution of cognitive functions among humans and machines. This methodology enhances user-centered and participatory design, and traceability of design decisions. It was successfully tested on three main applications in the aeronautics domain. One of them is presented.

Keywords

Active documents, aeronautics, evaluation, function allocation, automation, organizational memory systems, participatory design, safety critical systems.

INTRODUCTION

Automation has been mostly constructed and applied by engineers in the past. Human factors people have brought new principles and methods to test the usability of complex systems during the design process. However, end users are only one concern. Automation needs to be considered in a broader sense than just user-centered automation [1] because it should be done for the benefit of the largest range of people including users, designers, support people and trainers. Participatory design and traceability of design decisions (design history) is consequently a crucial issue [7], in particular, for the design and management of safety-critical systems. Safety-critical systems include, for example, critical-care, nuclear, emergency, military and aerospace systems. They are characterized by the following list of non-exhaustive issues: time-pressure, complexity, risk assessment and human reliability. A wrong function allocation in such systems may result in catastrophic accidents. The paper will first introduce an agent-oriented cognitive engineering model. Subsequently, the concept of cognitive function will be developed. Human-centered automation will be described in terms of cognitive function allocation among humans and machines along four fundamental dimensions: task, artifact, user and environment. Active design documents supporting the Cognitive Function Analysis (CFA) will be presented and illustrated as mediating tools that support the allocation process. An aeronautical example will illustrate the use of CFA. Conclusions and perspectives will be given in the balance of the paper.

CHI 98 Los Angeles CA USA

AGENT ORIENTATION: A COGNITIVE ENGINEERING MODEL

Both human and machine agents have cognitive functions

Safety-critical systems such as aircraft currently include a tremendous amount of computer software. Previous human-machine interaction that was energy-intensive has now evolved towards human-computer interaction that is information-intensive. The nature of interaction is quite different to the point that new jobs (represented by specific cognitive functions) have emerged. An aircraft pilot has become a manager of what can best be thought of as artificial agents. He or she needs to coordinate, trust, supervise and cooperate with these agents. Several traditional human factors principles and approaches have also become obsolete because the paradigm of a single agent, as an information processor, is no longer appropriate. Multi-agent models [13] are better suited to capture the essence of today's information-intensive safety-critical systems. A human agent interacting with a software agent [5] 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 [25]. Agent-to-agent communication has been described by many authors working in the domain of highly automated systems [1, 16]. Several attributes were used to describe automation. Among them, in addition to basic usability principles [17], and from our experience in aeronautics, the following were found important in multi-agent human-machine communication:

- 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;
- expertise-intensive versus common-sense interaction;
- programmability, i.e., ability to program and re-program highly automated systems.

The AUTO pyramid

An artifact is a physical or conceptual human-designed entity useful for a given class of users to perform specific tasks. Carroll and Rosson discussed transactions between tasks and artifacts in the human-computer interaction world [8]. It is sometimes very difficult to know if the task defines the artifact or if the artifact defines the task. In reality, users' profiles, tasks and artifacts are incrementally defined to satisfy a specific objective. The task and the user are usually taken into account implicitly. Task can be modeled from a task analysis or a model of the process that the artifact will help to perform. A specified task leads to a set of information requirements for the artifact. Conversely, the artifact sends back its own technological limitations according to the current availability of technology. Users can be incrementally taken into account in the design loop either through the development of descriptive or analogous user models. User modeling can be implicit or explicit, and leads to the definition of appropriate user profiles. When a version of the artifact and the task are available, a user can use the artifact to perform the task. An analysis of the user activity is then possible, which contributes to adapt both the task, procedures and training, and artifact ergonomics. The artifact-user-task triangle [4] implicitly defines an incremental approach to design/evaluation that is similar to the spiral model for software development [2].

Artifact design and use are defined not only from a local ergonomics viewpoint, but also from management and organizational viewpoints both in the short term and the long term. Global ergonomics expends the triangle to a pyramid by introducing a fourth concept: organizational environment. The *artifact-user-task-organizational environment* (AUTO) pyramid introduces three additional issues (Figure 1): the designed artifact emerges in the environment, and the environment evolves from the integration of the artifact; the task requires the organization of new jobs, and the environment sends back new roles; users using the artifact to perform the task in the organizational environment determine social issues.

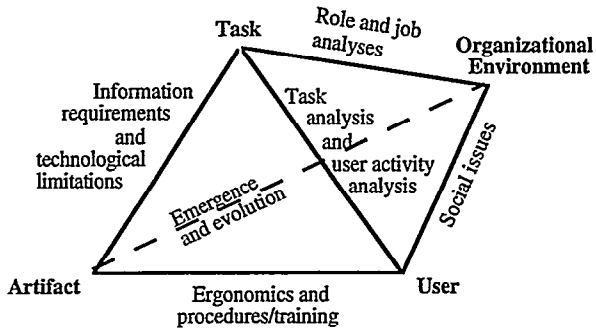


Figure 1. The AUTO pyramid.

Cognitive function definition

In highly dynamic complex automated systems, users develop cognitive skilled processes that are very context-sensitive. These numerous skills can be approximated by cognitive functions. By definition, a cognitive function enables its user to transform a (prescribed) task into an activity (effective task). It represents a human cognitive process that has a role in a limited context using a set of resources. The role of a cognitive function covers the concept of responsibility (who is in charge?) Eliciting a cognitive function requires one to specify its context of use (where and when this function is relevant and usable?) Unlike goal-driven models, such as GOMS [6], that tend to valorize smaller numbers of methods, context-driven models such as cognitive functions try to elicit organization of context patterns that facilitate the access to the right cognitive function at the right time. Cognitive functions are incrementally categorized according to context. A cognitive function is implementable when it is linked to right resources that are cognitive functions themselves. With respect to the AUTO pyramid (Figure 2), these resources can be user-based (e.g., knowledge and skills), task-based (e.g., checklists or procedures), artifact-based (e.g., artifact level of affordance [18]) or organizational environment-based (e.g., delegation to other agents).

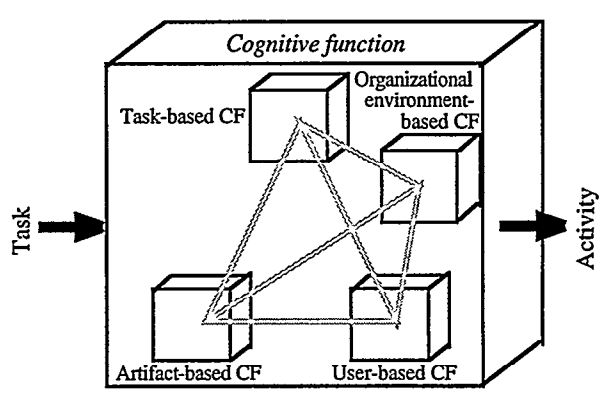


Figure 2. Four types of cognitive function components.

An important issue is to make the constraints explicit enough to guide the decisions during the design process. Cognitive functions are experimentally elicited by interpreting the deviations between the task and user activity in terms of role, context and resources. Examples of high-level cognitive functions are: situation identification, decision making, planning, and actions coordination.

HUMAN-CENTERED AUTOMATION OF SAFETY-CRITICAL SYSTEMS

The procedure-interface duality

In safety-critical systems, operational procedures are used in either normal or abnormal situations. 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, 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 [9]. In other words, people are not necessarily good procedure followers. They want to be in control [1]. 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 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.

Example of advanced cockpit automation

Prior to the integration of flight management computers (FMCs) onboard aircraft, pilots planned their flights using paper and pencil technology. An FMC 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 FMC drives the airplane by providing setpoints to the autopilot. The FMC computes the aircraft position continually, using stored aircraft performance data and navigation data [11]. The same kind of example was studied by Irving et al. using the GOMS approach [14], and experimentally by Sarter and Woods to study pilots' mental load model and awareness of the FMC [19]. An analysis of the cognitive functions involved in the use of the Multifunction Command and Display Unit (MCDU), the user interface of the FMC, enabled us to elicit a set of cognitive functions categorized according to the AUTO pyramid.

Programming a flight plan using a MCDU is a complex cognitive function that may be decomposed into several layers of simpler cognitive functions. Only task-based and artifact-based cognitive functions are elicited first. User-based and organizational environment-based cognitive functions are subsequently added to describe user's assets and problems as well as environmental issues. For instance, the *Preflight* task-based cognitive function is decomposed into three task-based cognitive functions *Setting up*, *Flight plan preparation*, and *Performance*. *Setting up* is then decomposed into two task-based cognitive functions *System status check*, and *Navaids deselection*. *System status check* is conditionally (*If A/C Status page is not displayed*) decomposed into four artifact-based cognitive functions *Depress 'DATA' key*, *Select 'A/C STATUS'*, *Check Database period of validity*, and *Check Clock/Date*. Anytime a cognitive function is elicited, its role and context of use are also described or refined.

An easy-to-use user interface usually results in affordable artifact-based cognitive functions. Most pilots find the MCDU difficult to learn and use. This complexity of use can be illustrated using two kinds of observations. First, the pilot needs to push keys, browse menus that can be more or less complicated due to the depth or recursion of these menus, i.e., artifact-based cognitive functions are often complicated. Second, pilots delegate complex cognitive functions, such as minimizing the distance between two geographical waypoints or satisfying a constraint imposed by air traffic control, to onboard computers that help manage the flight, i.e., task-based cognitive functions delegated to the machine are complex and require the involvement of information-intensive cognitive functions such as situation awareness and supervisory control [20].

Cognitive function allocation

The first step of CFA involves eliciting, constructing and chaining cognitive functions that are involved in a specific task. A second step involves a set of principles and guidelines that guide cognitive function allocation among agents, and help understand the repercussions of this allocation. These repercussions can be expressed in terms of new cognitive functions created and new relations between agents. The development of new tools can facilitate the execution of such cognitive functions by taking over part of the job currently done by humans.

Formalizing cognitive function allocation is a means of better understanding and controlling automation according to a list of automation goals such as those proposed by Billings for the air transportation system [1]: *safety*: to conduct all operations, from start to end, without harm to persons or property; *reliability*: to provide reliable operations without interference from environmental variables; *economy*: to conduct all operations as economically as possible; and *comfort*: to conduct all operations in a manner that maximizes users' and related persons' health and comfort. Human-centered automation principles should be clearly defined, e.g., technology-mediated human-human communication can be greatly enhanced by directing tedious and time-consuming cognitive functions towards the machine, and cognitive functions that keep user awareness and control of the situation towards the human.

CFA provides a theoretical basis supporting a current debate on direct manipulation versus interface agents [21]. Artifact-based cognitive function transfer from the user to the machine usually defines an automation that enhances direct manipulation. Task-based cognitive function transfer from the user to the machine defines an automation that enhances task delegation to a software agent. It distributes the responsibility of the task between the human and the machine. The way task-based cognitive function transfer is understood by designers is crucial because it defines the user's role and context of use of the machine. This is why a careful CFA is required to define roles, context of use and resources of each cognitive function involved in the human-machine interaction of safety-critical systems. For instance, it is often crucial that users perceive the level of autonomy of the designed artifact. The result is that the context of use of a cognitive function must be incrementally co-constructed by both designers and users in a participatory design framework that is proposed in the next section.

ACTIVE DESIGN DOCUMENT SUPPORT

Active design document definition

CFA is supported by a cooperative use of active design documents. Exploiting the procedure-interface duality issue and the AUTO pyramid, an active design document is defined by three aspects [3]:

- *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*;
- *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.*

Development of active design documents

After a first active design document is designed and developed (interface objects and interaction descriptions), a first round of analysis determines the first contextual links. Such an analysis is based on the evaluation of observed or reported human-machine interactions produced by typical users. An active design document can be refined either by: revising interaction descriptions under the requirements of previously generated contextual links and possibly the modification of interface objects; modifying interface objects under the requirements of previously generated contextual links and possibly the modification of interaction descriptions; or generating contextual links to provide information on flaws and relevant comments of the congruence between interaction descriptions and interface objects. Active design document creation and refinement is guided using usability principles and criteria that are based on domain requirements. In particular, contextual links are generated and structured according to these usability principles and criteria. They can be generated as: free text, quantitative and qualitative evaluations based on specific criteria and constraints.

Evaluation using cognitive function descriptors

Measuring is evaluating. A measurement is always based on a model or a theory. It can be subjective or objective according to the confidence that we have in both the model and the measurer. For a long time, human and social sciences implicitly acknowledged that quantitative measures were good (objective) evaluation tools. Unfortunately, quantification works on a closed-world and do not take into account unanticipated events very well. Thus, there was a need for a new type of approach. The expert system approach has revealed a new type of model based on the use of qualitative expertise. Instead of having a specified metrics, e.g. metrics in statistics, a few domain experts are required to provide their knowledge. Experts or key informants are usually good evaluators when they are provided with the right things to evaluate. They are also able to extend the initial set of criteria. This approach is thus more open-world and enables evaluation to take into account unanticipated events. Its weakness is that experts are subjective, based on their background, experience, skills and situation awareness. The choice of a (small) number of evaluators is thus crucial. It is guided by a good mix of common sense and domain knowledge.

The description of a cognitive function by a domain expert is often a good measure of the quality, relevance and usability of an artifact. Cognitive functions are elicited with respect to their role, context and associated resources. They are described by cognitive function descriptors (CFDs) that are measurable attributes constructed from domain knowledge and usability attributes of multi-agent human-

machine environments (already provided in this paper). CFDs should be clearly defined in order to be further compared and widely accepted. In other words, a CFD has the following properties: two CFDs provided by two experts A and B should be comparable, i.e., expressed properties should be clearly defined and consistent in both CFDs; this is a comparison issue; any CFD template should be defined according to current cognitive engineering results and the terminology of the application domain; this is a standardization issue. CFDs are usually defined from observation in work situations and in meetings of experts. The following CFDs constitute a potential list of usability criteria for HCA in the MCDU/FMC domain: *long-term memory* (LTM): necessary effort to recall the way to execute an instruction; *data affordance* (DA): information relevance and capacity to guide the user on the next action to perform; explicit data enable the user to minimize his or her memory effort and workload before starting the next action; *data readability* (DR): data representation, format and font; *feedback* (FB): system reaction after each action of the user; *data format* (DF): consistency between data format and insertion-identification procedure; *error tolerance* (ET): human error possibility, importance of its consequence and its recovery ease; *keystroke number* (KN): number of keystrokes to perform an action; *recursion levels* (RL): number of recursion levels to perform an action. In addition, each qualitative CFD is typically evaluated according to a five-value scale: 1: excellent; 2: good; 3: medium; 4: poor; 5: unacceptable. An example of evaluation results included in a contextual link of an active design document is provided in Table 1.

Cognitive Function Level 1	Cognitive Function Level 2	LTM	DA	DR	FB	DF	ET	KN	RL
WIND ENTRY: format error	- if F-PLN page not displayed DEPRESS F-PLN key	3	2		1			1	+1
	- DEPRESS 'NEXT PAGE' key	5	3		1			1	+1
	- SELECT 'VERT REV WPT'	3	2	4	1			1	+1
	- ENTER & INSERT 'WIND'	1	2	4	1	3		8	
	Total						4	**11**	**3**
RECOVERY									
- erase message 'FORMAT ERROR'	- DEPRESS 'CLR' key	5	3		1			1	
- erase invalid data	- DEPRESS 'CLR' key 5 times	3	2		1			5	
- enter corrected data	- ENTER & INSERT 'CORRECT WIND'	2	1	4	1	3	5	5	
	Total							**11**	

Table 1. An example of CFA results for a classical MCDU.

Incremental generation of active design documents

In the FMC experiment, the general trend was to move from a generic MCDU interface, e.g., including generic functions keys, to an integrated interface that includes affordable interface objects. These affordable interface objects are hypermedia objects that can be easily modified during the design process, and include relevant properties and behaviors that are specified from the first CFA results. Figure 3 presents an example of an alternative interface for programming the FMC. In this kind of interface, waypoints and trajectories are interface objects that have properties and behaviors. For instance, the preprogrammed waypoint TRS1 can be changed into the waypoint TRS2 by simply selecting it. When TRS2 is selected, the trajectory is automatically redrawn. Main advantages of interface objects direct manipulation are:

- quick access to the appropriate information;
- easy understanding of what to do (i.e., natural interaction with interface objects);
- immediate feedback, visualization of usual objects, and affordance to assess them against expected results.

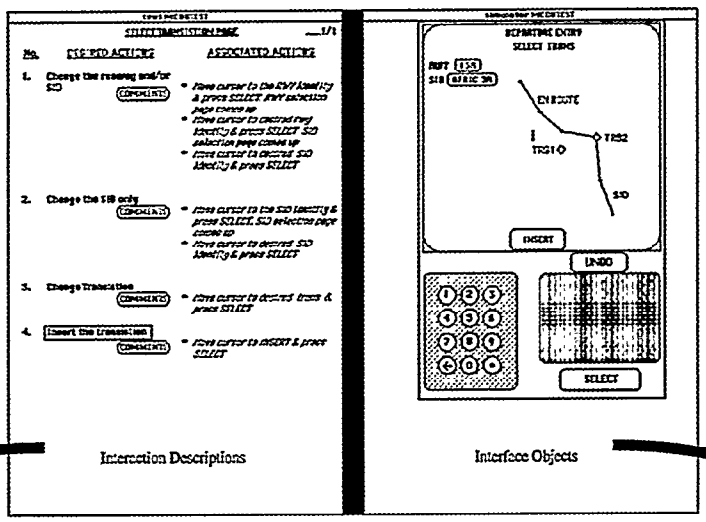
In other words, the pilot does not have to search for waypoints by browsing FMC pages using a classical MCDU. He or she directly manipulates meaningful objects that appropriately react to provide immediate possible configurations. The example provided in Figure 3 shows that a new interaction device is necessary to manipulate interface objects such as waypoints. The trackpad was chosen for environmental reasons. A second cognitive function analysis was then performed. Results are presented in the form of tables (Figure 3). The first observation of this table shows that scores are closer to 1 than to 5,... as expected!

Participatory design and traceability issues

Active design document generation and maintenance concretizes Muller's arguments in favor of *participatory design* [16]: to combine diverse sources of expertise; to formalize the ownership and commitment by all of the people who will eventually work on or with the designed artifact; to participate in decision-making by the people who will be affected by the design decisions. Active design documents are shareable prototypes of the real artifacts being designed that can be used by real users to assess their usability. Prototypes should be familiar to users. Their limitations should be clearly identified. A shareable prototype should be understandable by all the members of the design team and keep them on a common track. Active design documents enable the design team to share concepts by writing and reading their expression in the form of multimedia objects. They are incrementally modified to harmonize mutual understanding of design team members. The basic difference between the classical human-factors-oriented design and participatory design is that instead of analyzing the existing user organization and the application

area, design team members learn from each other. Active design documents define an active *external memory*. They are incrementally modified according to possible design options, human factors evaluation criteria and organizational requirements. In the CFA approach, modifications are induced from interaction among design team members. Active design documents may come to dead ends as well as evolve into currently valid documents describing the artifact. Corresponding document evolution is kept to preserve design history.

Basically, in the beginning of the design process, active design documents may have large interaction descriptions that document a preliminary task analysis, roughly sketched interface objects, and contextual links mainly defined by early design rationale. Later in the life cycle of the artifact, active design documents interface objects become more sophisticated and user-friendly, interaction descriptions should become minimal, and contextual links richer in comments and feedback from tests. The shorter and crisper interaction descriptions are, the easier the interaction with interface objects is. An important issue is to handle the growth of contextual links. This is precisely where traceability problems arise. We call *traceability* the process that enables one to recall design decisions and the various alternatives as well as why these alternatives were not chosen. Contextual links are used to implement an indexing mechanism. They should be classified, generalized, and incrementally simplified in order to be efficiently used. A first solution is to group them by viewpoint.



Cognitive Function Level 1	Cognitive Function Level 2		LTM	DA	DR	FB	DF	ET	KN	GF	RL
WIND ENTRY: format error	- if F-PLN page not displayed										
	RETURN TO CURRENT PAGE		3	1	1	1			1	1	
	- SELECT VERTICAL WIND'		3	1	1	1			1	2	+1
	- CHECK/MODIFY 'WIND'		3	1	1	1	1	1	4	2	
		Total							**6**	**5**	**1**
RECOVERY											
- acknowledge message (click OK)	- DEPRESS 'SELECT'				1			1	1		
- enter corrected data	- ENTER & INSERT CORRECT WIND'		3	1	1	1	1	1	7	2	
		Total							**8**	**2**	

Figure 3. Example of a MCDU alternative interface, the associated procedure (interaction description) and the content of a contextual link expressed in the form of a CFD table (the whole imbedded in an active document).

RELATED WORK

Depending on the type of behavior, two types of analysis are possible:

- A goal-oriented task analysis involves a hierarchic decomposition of goals into subgoals, and so on until basic actions are found and executed. The corresponding scientific approach is top-down, based on analytical descriptions. It usually attempts to model internal cognitive mechanisms of a single agent, and to describe exhaustively the goal space.
- An event-oriented task analysis involves an incremental composition of events into sub-contexts and contexts. The corresponding scientific approach is bottom-up, based on observation protocols of situated actions [22]. It usually attempts to model multi-agent interaction within an organizational environment, and to describe exhaustively the context space.

When a human performs a task, his or her behavior is opportunistic, i.e., both intentional and reactive. In the control of complex dynamic systems, human operators need to be and are opportunistic. They need to be ahead of the machine (goal-driven anticipation) and respond quickly to events that are not anticipated (event-driven reaction). With

respect to some typical events they may change their strategies, i.e., their goal-driven behavior. CFA differs from GOMS techniques [6, 14] because it attempts to incrementally model contexts of use of both human and machine cognitive functions. Since safety-critical systems such as aircraft induce both intentional and reactive behaviors, CFA is very appropriate to study and describe situation awareness, human errors, cooperation, coordination, for instance.

	Goal-oriented task analysis	Event-oriented task analysis
Human behavior models	Intentional and deliberative	Reactive and explicative
Approach	Top-down based on analytical descriptions	Bottom-up based on cooperative observation protocols and interactions with users
Modeled process	Internal model of an agent	Interaction between several agents
Goal space	Strongly defined, limited by the granularity of the description	Loosely defined
Context space	Loosely defined	Strongly defined, limited by the granularity of the description

Table 2. Goal- versus event-oriented task analysis.

Table 2 presents the advantages and drawbacks of these two different task analysis approaches. Goal-driven approaches are well adapted to analyze problem solving. Event-driven approaches are better suited to analyze problem formulation or problem setting (situation patterns) in complex system environments. Indeed, a problem is characterized by a problem statement and a problem solving process leading to a solution. Everybody knows that a well stated problem is already half solved and this is well adapted to (successful) reactive behavior. Moreover, when a beginner starts to learn a particular domain, he or she starts learning problem solving methods (analytical knowledge) which he or she will improve incrementally simply by augmenting and improving these methods, and also by improving the way he or she formulates problems. In a cognitive function analysis, the emphasis is put more on problem formulation, and then context, than on problem solving. Problem formulation, like problem solving, is an incremental process that again calls for Pasteur's 'prepared mind.' But it is not a magical or mysterious process [15]. CFA combines both goal- and event-oriented analyses within a single framework uspported by active design documents.

CFA is a global approach that considers the task as part of an overall framework that also includes the artifact, the user and the organizational environment. In other words, the task cannot be isolated from the actual work that includes a description of three types of contraints, i.e., roles, contexts and involved resources. Similarly, Vicente and Pejtersen [24] propose a constraint-based approach to work analysis (that focuses on flexibility and broad scope of applicability), instead of an instruction-based approach to task analysis (that focuses on efficiency of task performance). A constraint-based approach does not tell you the right way to do your task. It just lists constraints. As in CFA, this approach leads also to a functional description for human-machine systems.

Active design documents support sketching [12] as mediating tools for design team members. They also enable one to trace design decisions based on the evaluation of cognitive function descriptors. From this perspective, CFA has similarities to *Raison d'Etre* [9]. CFA contributes to the creation and maintenance of a living design memory [23].

CONCLUSIONS AND PERSPECTIVES

This paper has presented a methodology for human-centered design of highly automated safety-critical systems. It focusses on cognitive function allocation using a combined analytical and situated (empirical) methodology to human-centered automation.

CFA enables the investigator to describe cognitive functions with respect to the constraints and limitations imposed by the artifact, the user, the task, and the organizational environment.

CFA attacks the difficult issue of function allocation. It enables the description of how new technology influences distributed cognition by using a participatory design tool both mediating creativity and evaluation, and accounting for design history. Since it is very difficult and sometimes impossible to predict design-induced errors that lead to incidents or accidents, incremental evaluations and refinements are mandatory during the overall life cycle of an artifact. Active design documents offer the opportunity to users and other parties involved in the life-cycle of an artifact to formally influence its design. CFA supported by the effective use of active design documents provides descriptions of possible interaction, design rationale and evaluations linked to the actual artifact.

By enabling the visualization of interaction descriptions, interface objects and cognitive functions involved in the use of the artifact being designed, the design team is likely to anticipate and discover more contexts of use, more appropriate resources to perform the task and cooperative features required within the organizational environment. Since automation always leads to the definition of new roles, and possibly jobs for users, CFA offers a framework to elicit and analyze these new roles and changes. In particular, CFA is useful to analyze and possibly anticipate new risks in safety-critical systems.

The traceability of design rationale and associated human-factors-oriented evaluations represents a real asset for the organization that develops an artifact. Active design documents are designed and refined from the beginning to the end of the artifact life-cycle. A remaining important issue is to justify time and money spent in the implementation of CFA in a large-size industrial organization. Estimated development costs should be compared to the costs of late modifications of the artifact, incidents and accidents due to design flaws, and unnecessary training or maintenance. An evaluation framework, such as proposed by Zimmermann and Selvin [26], should be set up to assess the methodology against organizational requirements and current needs.

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. Jonathan Grudin, David Novick, Helen Wilson and anonymous reviewers provided astute advice towards improving the quality of this paper.

REFERENCES

1. Billings, C.E. (1991). *Human-centered aircraft automation philosphy*. NASA TM 103885, NASA Ames Research Center, Moffett Field, CA, USA
2. Boehm, B.W. (1988). A spiral model of software development and enhancement. *IEEE Computer*, Vol. 21, no. 5, May, pp. 61-72.
3. Boy, G.A. (1997). Active design documents. *Proceedings of ACM DIS'97 Conference*. ACM Press, New York.
4. Boy, G.A. (1998). *Cognitive Function Analysis*. Ablex Publishing Corporation, Greenwich, CT.
5. Bradshaw, J. (1997). *Software Agents*. MIT/AAAI Press, Cambridge, MA, USA.
6. Card, S.K., Moran, T.P. & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum.
7. Carroll, J.M. and Moran, T. (1991). Introduction to this special issue on design rationale. *Human-Computer Interaction*, Lawrence Erlbaum Associate, Inc., Vol. 6.
8. Carroll, J.M. & Rosson, M.B. (1991). Deliberated Evolution: Stalking the View Matcher in Design Space. *Human-Computer Interaction*, Lawrence Erlbaum Associate, Inc., Volume 6, pp. 281-318.
9. Carroll, J.M., Alpert, S.R., Karat, J., Van Deusen, M.S. & Rosson, M.B. (1994). Raison d'Etre: Capturing design history and rationale in multimedia narratives. *Proceedings of the ACM CHI'94 Conference*. (Boston, April 24-28). New York: ACM Press, pp. 192-197, 478.
10. De Brito, G., Pinet, J. & Boy, G. (1997). *Checklist use in new generation aircraft*. EURISCO Technical Report no. T-97-042.
11. FCOM-A320 (1997). *Flight Crew Operation Manual A320*. Airbus Industrie, Toulouse-Blagnac, France.
12. Fischer, G., Nakakoji, K. & Oswald, J. (1995). Supporting the evolution of design artifacts with representations of context and intent. *Proceedings of Designing Interactive Systems (DIS'95)*. ACM Press, pp. 7-15.
13. Hutchins, E. (1995). How a Cockpit Remembers its Speeds. *Cognitive Science*, 19, pp. 265-288.
14. 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, pp. 344-350.
15. Langley, P.W., Simon, H.A., Bradshaw, G. & Zytkow, J.M. (1987). *Scientific dicovery; An account of the creative process*. MIT Press. Boston, MA.
16. 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.
17. Nielsen, J. (1993). *Usability engineering*. Academic Press. London.
18. Norman, D.A. (1992). Turn Signals are the Facial Expressions of Automobiles. Reading, MA: Addison-Wesley.
19. Sarter, N.B. & Woods, D.D. (1994). Pilot interaction with cockpit automation II: An experimental study of pilots' mental model and awareness of the Flight Management System. *International Journal of Aviation Psychology*, 4, pp. 1-28.
20. Sheridan, T.B. (1984). Supervisory control of remote manipulators, vehicules and dynamic processes: experiment in command and display aiding. *Advances in Man-Machine Systems Research*, J.A.I. Press, 1, pp. 49-137.
21. Shneiderman, B. & Maes, P. (1997). Direct manipulation vs. Interface agents. *interactions*, November-December, volume IV.6.
22. Suchman, L.A. (1987). *Plans and situated actions. The problem of human-machine communication*. Cambridge, England: Cambridge University Press
23. Terveeen, L.G, Selfridge, P.G. & Long, M.D. (1995). Living design memory: Framework, implementation and lesson learned. *Human-Computer Interaction*, Volume 10, pp. 1-37.
24. Vicente, K. J., & Pejtersen, A. M. (in preparation). *Cognitive work analysis: Towards, safe, productive, and healthy computer-based work*. Mahwah, NJ: Erlbaum.
25. Wiener, E.L. (1995). The focus of automation: Which tasks should be automated. *EURISCO Industrial Summer School on Human-Centered Automation*. Saint-Lary, France.
26. Zimmermann, B. & Selvin, A.M. (1997). A framework for assessing group memory approaches for software design projects. *Proceedings of DIS'97*. August 18-20, Amsterdam, The Netherlands.