Active Design Documents

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

Technical documents are created, modified and used during the life cycle of an artifact. They can be more or less formal, ranging from normative knowledge-based representations to natural language. They are also tools that support dialogue between designers, manufacturers, trainers, legislators and users. Active design documents (ADDs) are a new generation of support for cooperative work of design teams. ADDs include interaction descriptions (Ids) that provide the way the artifact should be used, interface objects (IOs) that provide an interactive prototype of the artifact, and contextual links (CLs) that enable the storage of evaluations and explanations of the distance between IDs and IOs. Incremental ADD design and evaluation contribute to instantiate a participatory design process and a formal trace of the design rationale as a function of usability criteria. An application in the aeronautics domain is presented.

KEYWORDS

Active documents, hypertext, participatory design, evaluation.

INTRODUCTION

In the classical cycle of technical documentation, design teams write requirement documents for manufacturing teams who then write documents for users. Design documents usually describe the way the artifact works. Operational documentation is usually developed at the end of the artifact development process. It unfortunately often attempts to compensate for design flaws.

This paper proposes a different perspective. It presents the motivations of the approach, the definition of an active design document, related functionality in terms of cognitive functions involved in the interaction and the issue of traceability of design decisions.

WHY ACTIVE DESIGN DOCUMENTS?

Design teams exchange vivid knowledge of artifacts that they develop. For instance, design team players talk about the artifact, reinforce ideas, disagree with each other, or reach consensus. Descriptions and arguments remain traditionally documented in the form of text and drawings. Hypertext linking between technical documentation and artifacts provides a more active description of the ways in which the artifact works or should be used. Resulting documents enable the description of how the artifact works and how it should be used. In addition, linking interaction descriptions to corresponding artifact functions (Figure 1) is a step towards the formalization of cognitive functions involved in the use of the artifact.

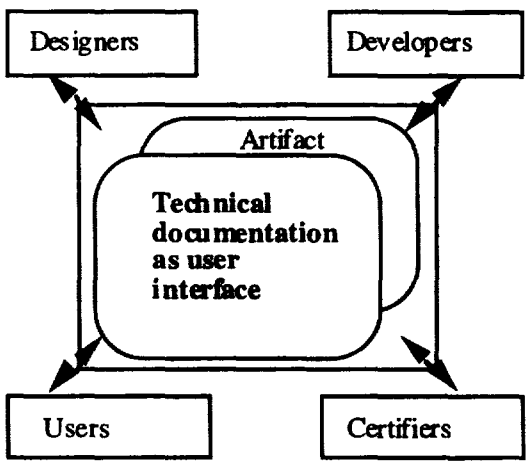

Figure 1. Technical documentation as a user interface of the artifact.

A cognitive function can be defined both mathematically as a transformation of a task into an activity, and teleologically as an agent with a role defined in a limited context using a set of resources. Cognitive function elicitation enables the design of interface objects that afford direct manipulation [2] [4] [7]. The cognitive function analysis (CFA) methodology [4] and more specifically the distribution of these functions between human and artifact are based on the assumption that the artifact behavior induces user attitudes. The user reacts to artifact behavior and constructs his or her own attitudes to avoid cognitive dissonance [9]. Cognitive dissonance results from bad pattern matching between user expectations and real possibilities that the artifact can afford. User attitudes are based on a sum of beliefs and

implicit evaluations of use possibilities. These attitudes lead to intent and behavior formation.

« Writing is design, design is writing » [12]

Our approach supports the thesis that the quality of technical documentation contributes to the quality of design. We usually write for potential readers. In the same way, we design for potential users. We know that papers that we write must be reviewed by several persons before being delivered outside. We also know that artifacts must be tested by several persons before being delivered outside. The reader of a multimedia document has become a user of a software application. From this viewpoint, reading has evolved towards human-computer interaction (HCI). Writing has also evolved towards design of interactive software. Writing words, phrases, paragraphs and chapters has become designing objects and software agents [6]. Static paper documents have become (inter)active documents.

The active part of a book (system) is the reader (user). In addition, the organization of the book (system), the way phrases (objects) are written (designed), used style and lexicon used suggest reader (user) activity. Sometimes, the reader (user) hardly understands what the author (designer) wanted to express. Instead of mobilizing reader (user) cognition on interaction problems, the most important part of the cognitive activity of the reader (user) should be centered on the understanding and interpretation of (active) document content.

The need for contextualization shapes human-centered design

Human-centered design methods take into account users' needs and requirements in the design/evaluation process. Instead of designing an artifact and documenting it later, we design and evaluate documented prototypes, called active design documents (ADD) in this paper, incrementally until they become acceptable prototypes. A main difficulty in technical document design is to anticipate a very large number of contexts of use. Context of use is related to entities such as situations, behavior, viewpoints, and dialogue. Conventional paper technology is not an appropriate support for contextualization. Software technology provides more contextualization capabilities. In addition, contextualization is both an intra-document as well as an inter-documents issue, for support of traceability.

ACTIVE DESIGN DOCUMENT: DEFINITION

An ADD is a hypermedia application usable by a community of persons. An ADD describes various attributes of an artifact (being or actually designed). An ADD is defined by three aspects:

- *interaction descriptions* (IDs) constitute the task space – they describe how to use the artifact, e.g., a procedure to follow (left part of Figure 2);
- *interface objets* (IOs) connected to IDs constitute the activity space — they enable the user to actually use a simulation of the artifact, e.g., a pilot can test a software prototype of a flight management system (FMS) interface (right part of Figure 2);
- *contextual links* (CLs) between the IDs and the IOs constitute the cognitive function space – they enable the evaluator to annotate and comment the active design document during a usability test.

Interaction descriptions

Interaction descriptions (IDs) of an ADD constitute the core of the human-artifact dialogue requirements. IDs may be expressed either in natural language, or in a domain-specific language ranging from Simplified English to a knowledge representation such as knowledge blocks [4]. Knowledge block descriptions enable semi-formal analysis of interaction complexity [5], and elicitation of contexts of use as well as abnormal conditions. The use of knowledge blocks as a support for the generation of IDs is presented elsewhere [4]. In this paper, ADDs are textual descriptions, e.g., operational procedures (see Figure 2).

Interface objects

Interface objects (IOs) of an ADD provide an appropriate, useful and natural illusion of the artifact. IOs enable the user to interact. They include important dynamic aspects of artifact properties such as color changes with respect to specific semantics or continuous parameters evolution. A user following IDs and interacting with IOs is able to test artifact usability. For example, if the use of an artifact requires too much learning or is of little interest, it might not be used. IOs may constitute the final product or an intermediary prototype.

Comments as contextual links

An ADD is equipped with an evaluation support that includes indexing, annotating (evaluation history) and browsing (hypertextual traceability, i.e., relating IOs to IDs as well as ADDs among each other). The notion of context enables the customization and adaptation of multimedia documents to user requirements. In CID for instance [1], contextual links are incrementally generated by interpreting and annotating an integrated documentation. They are processed by a machine learning mechanism. For example, IDs generated by an author-designer or by a reader-user are often not the same, even if the same person is a designer and a user. The CFA methodology [4] generalizes the CID approach: a contextual link base is incrementally generated by specifying, interpreting and annotating ADDs (Figure 3).

Interaction descriptions and interface objects are concrete implementations of descriptors that can be clickable strings or graphical areas (e.g., an instrument or a part of an instrument such as a speed indicator in an aircraft cockpit). When a user selects a descriptor, he or she obtains one or several referents. These referents are consistent windows including IDs or IOs. Contextual links (CLs) elicitation consists of providing viewpoints

on descriptor-referents relations. Either the user follows IDs and produces an activity by using corresponding IOs, or the user interacts directly with IOs and verifies the validity of corresponding IDs. CLs are generated and used incrementally to improve ADDs. They describe both design knowledge and artifact usability properties associated to contexts of use.

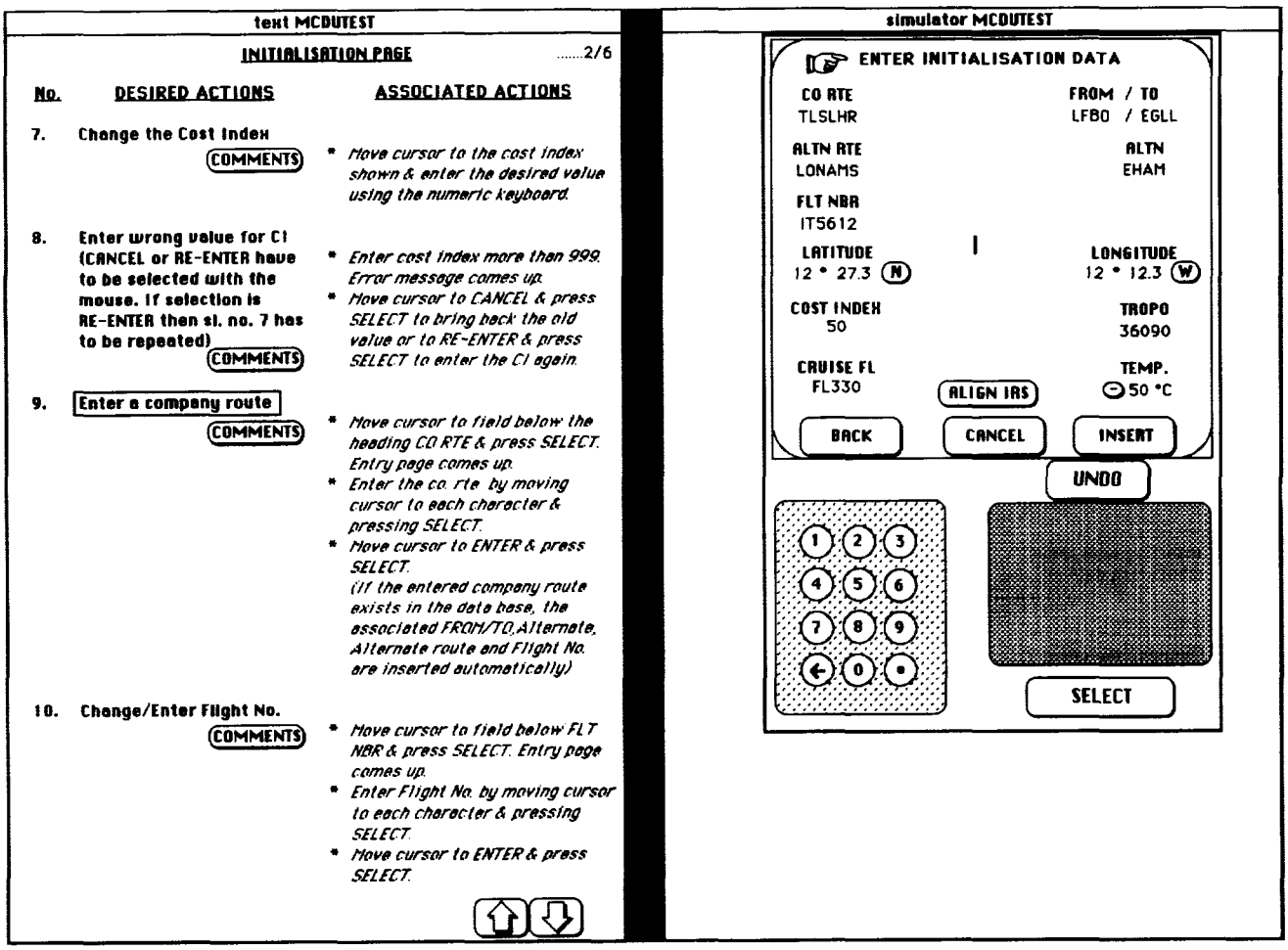

Figure 2. An active design document: textual descriptions of required interaction (left part) and interface objects (right part).

ACTIVE DESIGN DOCUMENTS AS A MEDIATING ENVIRONMENT FOR COGNITIVE FUNCTIONS ELICITATION

The basic claim of the CFA approach is that elicitation of cognitive functions involved in the interaction is equivalent to incrementally constructing a cognitive model of the use of the artifact, and to evaluating appropriate matching between interaction descriptions and interface objects.

Active design documents as a support to understanding and evaluation

Drawings are useful for explaining an idea or a concept. These drawings can be done on any kind of support including a restaurant's paper tablecloth or a blackboard. People usually choose cheap easy ways to discuss and formalize ideas. ADDs are intermediary supports between this type of communication and cooperation support (generally conventional paper documents), and fully dynamic simulators. If paper documents are cheap to generate, they are usually difficult to read and understand because they are not vivid enough. Simulations are much more expensive to generate, but can be very useful to test artifact usability. ADDs constitute a compromise between these two extremes. By using rapid prototyping tools, ADDs are easier to generate. They are interactive and enable global and local usability testing.

An active design document can be shallow-global or deep-local

The prediction power of an ADD depends on the level and nature of its details. The more people are able to relate an ADD to the real world, the more it enables believable evaluation. There is, however, a compromise to be made

between the level of necessary details as far as the development effort is concerned. Two types of ADDs can be distinguished:

- *shallow-global* ADDs that provide a global active view of the artifact; they do not have full deep functionality, but they offer the possibility of using supervisory control, management or coordination cognitive functions for instance;
- *deep-local* ADDs that provide a local active view of the artifact being designed; they have full deep functionality of the artifact for very narrow tasks, and offer the possibility of using control and monitoring cognitive functions for instance.

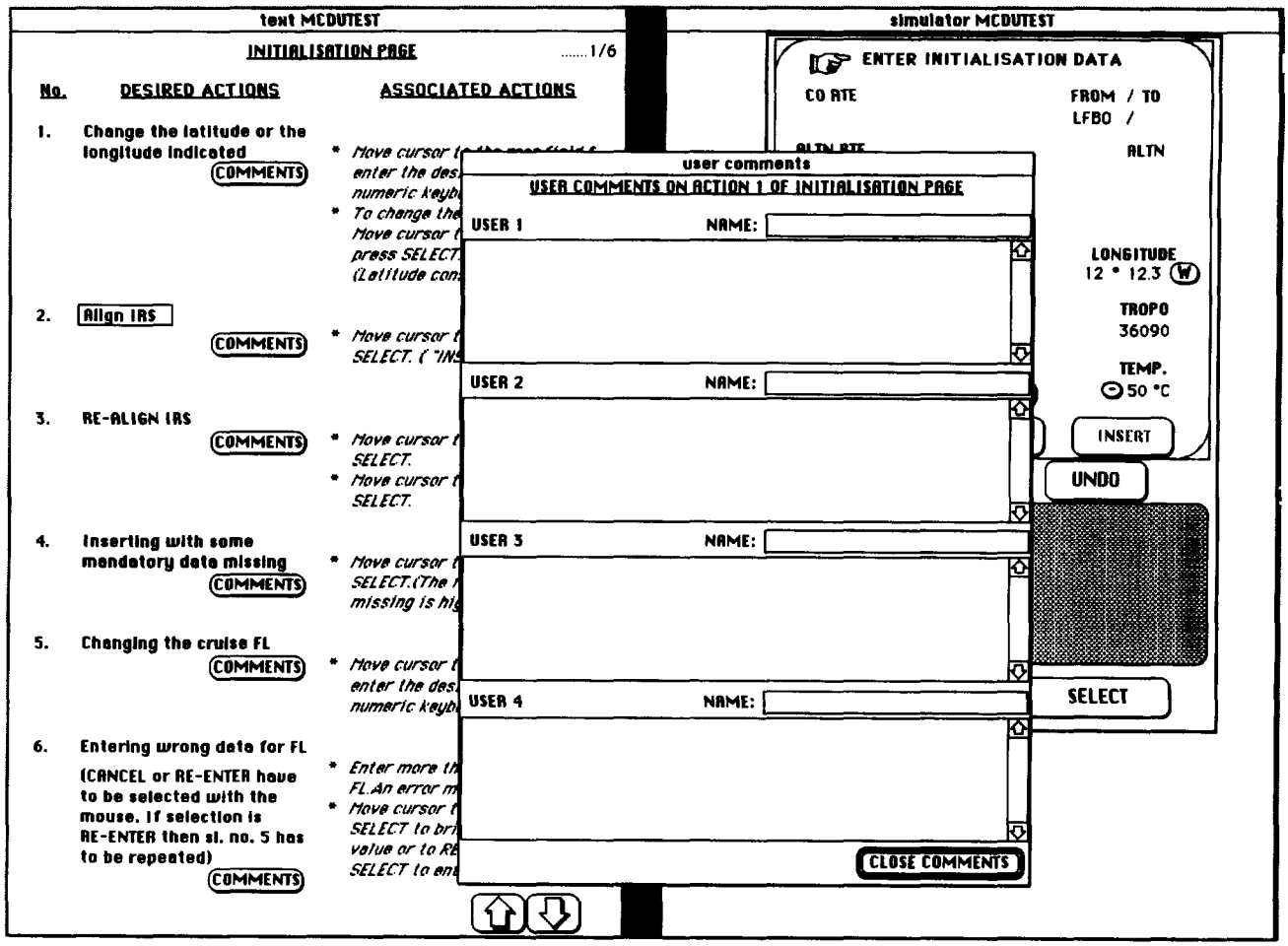

Figure 3. An example of comments as contextual links.

Cognitive function descriptors

Expert description of cognitive functions involved in the interaction is a way to test artifact usability. Cognitive function descriptors (CFDs) must be clearly defined to be compared and widely accepted. A CFD has the following properties: two CFDs provided by two different experts must be comparable (comparison property); and any CFD must be defined with respect to current scientific results in human-computer interaction and cognitive engineering, and domain terminology.

The following CFDs were used as usability criteria in human-centered automation: prediction (capacity of anticipation of action consequences in highly automated systems); feedback (quality and speed); autonomy (domain of artifact autonomous performance); elegance (artifact capacity to avoid additional inappropriate cognitive workload, essentially in critical situations); trust; qualification level (ranging from the need for expertise to an interaction based on common sense); and programmability.

The main issue is to understand if identified design flaws come from prototype approximations (shallow-global or deep-local) or from the developed concept itself. In the first case, it is a matter of explanation (e.g., wrong granularity, too shallow representation, bad assessment of the context of use). In the second case, flaw identification leads to revising design rationale.

AN EXAMPLE OF APPLICATION IN AERONAUTICS

Beginning in 1994, we have tested the use of ADDs on three types of applications in aeronautics: redesign of the user interface of a flight planning system onboard new-generation airplanes; design and evaluation of the user interface of an air-ground datalink communication system; and redesign of an electronic centralized aircraft monitoring system (ECAM) (example presented below). In these three cases, we directly faced industrial safety-

critical systems. Results were well received by industrial experts in aeronautics [2] [7] [10].

In these three applications, hypermedia technology was used to build ADDs. An ADD is constituted of three windows:

- the IDs window;
- the IOs window;
- the CLs window.

Figure 5 presents an example of an ADD for the Level Change procedure of a new generation commercial aircraft. This example has been developed using Hypercard software and its scripting language Hypertalk. Figure 5 presents an IDs window (right side) and an IOs window (left side). The user has already selected seven procedure items (they are highlighted). Each time a procedure item (i.e., an ID) is selected a contextual link automatically sends a message to the corresponding IOs window that produces an appropriate behavior. This behavior is evaluated and the corresponding contextual link is informed by generating relevant CFDs.

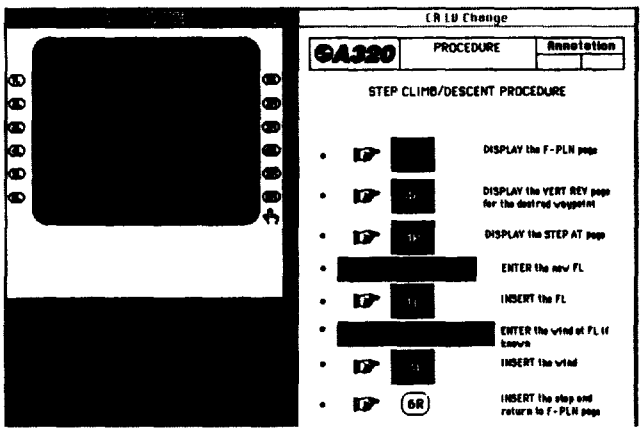

Figure 5. Example of an ADD of an Airbus procedure (IDs) and a simulation of an ECAM screen (IOs).

TRACEABILITY, DESIGN RATIONALE AND EXTERNAL MEMORY

ADDs are permanent records supporting communication between the actors involved in the life cycle of an artifact.

Active design documents and participatory design

ADD's generation and maintenance enable domain actors to share concepts by writing and reading them (in the multimedia sense), and to be part of the artifact design-use-evaluation spiral. This approach concretizes Muller's arguments advocating participatory design [11] :

- 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.

The main difference between classical human-factors-oriented design and this type of participatory design is that instead of simply analyzing the existing artifact life-cycle, actors train themselves by cooperating throughout ADDs. The first approach is based on observation, the second one is based on cooperation.

By providing users with design-aid tools such as ADDs, we enable them to contribute actively to design. Our first validation results (still preliminary) show this trend of social integration of users in the design process. It should be noted that the use of ADDs can be done in conjunction with methods such as the Group Elicitation Method (GEM) [3].

Evolution of active design documents and traceability

ADDs are continually modified with respect to opinions of various artifact life-cycle actors, evaluation criteria (CFDs for instance) and domain and cultural requirements. When the ADD evolution leads to dead-ends, a backtracking is performed to specific decisions that were made earlier, and a design history is kept in an external memory [8]. This approach reveals that indexing [1] is a crucial issue to enable the traceability of design decisions that are included both in ADDs and in the relations between ADDs (Figure 4).

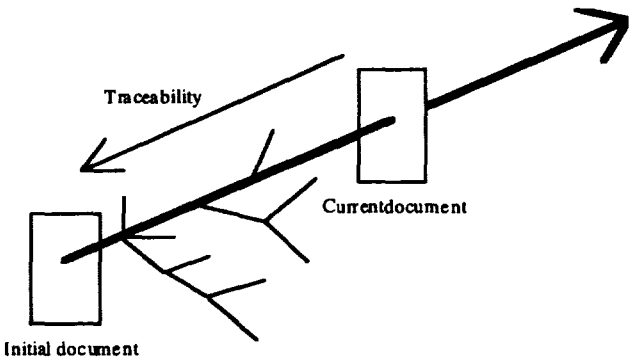

Figure 4. Evolution of a design document

The resulting library of ADDs is defined as an external memory. The active use of ADDs, i.e., not only reading but also writing, will contribute to change the organization of the designers'-users' space and will really define a human-centered design environment.

Basically, in the beginning of the design process ADDs include design-centered IDs that document a preliminary task analysis, roughly sketched IOs, and CLs mainly defined by the design rationale based on a first set of overall requirements. Later in the life cycle of the artifact, IOs become more sophisticated and user-friendly, IDs should become minimal, and CLs richer in comments and feedback from tests. The easier the interaction with IOs is, the shorter and crisper IDs are. An important issue involves how to handle the growth of contextual links. This is precisely where traceability problems arise. CLs should be classified, generalized and incrementally simplified (sometimes forgotten) in order to be used efficiently. A first solution is to group them by viewpoint.

CONCLUSION

This paper has presented an integrated approach to human-machine systems design by using active design documents that are not only an output of the design process but an active support to actors contributing to this process. ADDs are designed and refined from the beginning to the end of the artifact life-cycle. Design and use of ADDs enable the elicitation of interaction descriptions and interface objects. Contextual links enable design rationale traceability, and contribute to improving participatory design. Hypermedia technology is a good support to the construction and use of ADDs. This approach is currently used in three aeronautical projects for the design of safety-critical systems. In this type of projects, designers, pilots and human factors specialists cooperate in designing new cockpits by using ADDs. We envisage extending their use to training.

A remaining important issue is to justify time and money spent in the implementation of ADDs in a large-size industrial organization. We think that 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.

A possible extension of this approach consists of integrating GEM [3] in an electronic form using ADDs to guide brainwriting. ADDs could be exchanged within an Intranet supporting an integrated concurrent engineering approach.

ACKNOWLEDGMENTS

Many people contributed to the current state of EURISCO user-centered design using active documents project. Among them, I owe thanks to Hubert L'Ebraly, Thierry Broignez, Christophe Solans and Krishnakumar who greatly contributed to the development of the first versions of active technical documents for design. Meriem Chater, David Novick, and Helen Wilson provided astute advice towards improving the quality of this paper.

REFERENCES

1. Boy, G.A. (1991). Indexing Hypertext Documents in Context. *Proceedings of the Hypertext'91 Conference*, San Antonio, Texas, December.
2. Boy, G.A. & L'Ebraly, H. (1994). *Modèle de tâche et representation de l'expertise pour la conception de dispositifs d'interaction homme-machine. Exemple d'application en aéronautique.* Rapport EURISCO no. T-94-009-VI.
3. Boy, G.A. (1997). The Group Elicitation Method for Participatory Design and Usability Testing. Interactions. March-April issue, Vol. IV.2, pp. 27-33.
4. Boy, G.A. (1997). *Cognitive Function Analysis.* Ablex Publishing Corporation, Greenwich, CT. To appear.
5. Boy, G.A. & Bertuccio, E. (1997). Interaction complexity testing using the knowledge block representation. EURISCO publication to appear.
6. Bradshaw, J. (1997). *Software Agents.* MIT/AAAI Press, Cambridge, MA, USA.
7. Broigniez, T. (1996). *Application d'une méthode de conception interactive pour l'amélioration d'interfaces. Exemple d'application en aéronautique.* Rapport EURISCO no. RS-96-006-TB.
8. 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.
9. Festinger, L. (1957). *A theory of cognitive dissonance.* Evanston, Row, Peterson.
10. Krishnakumar, S. (1996). *Conception d'une interface homme-machine et d'un document actif en aéronautique embarquée.* Mémoire de Mastère TAS. ENSAE-EURISCO.
11. 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.
12. Norman, D.A. (1992). *Turn Signals are the Facial Expressions of Automobiles.* Reading, MA: Addison-Wesley.