ORGANIZATIONAL MEMORY 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: Organizational memory systems (OMS) are currently related to the development of Intranets that will enable massive information transfer within an organization. But they do not solve the major problem of existence or availability of the right information at the right time in the right place, and in the right understandable format. To this end, representing context is a crucial issue that facilitates appropriate information retrieval and understanding. In addition, organizations evolve towards more autonomy, cooperation and coordination between agents. Machine agents, taking the form of software assistants, tend to replace human agents in many tasks. In particular, a new kind of memory support using current information technology is presented: active documents. They are mediating tools that enable people to store and use both content and context. The emergence of these new agents is creating new human roles, and new jobs sometimes. This plenary paper presents and discusses the strengths and weaknesses of a variety of OMSs within industrial environments. Copyright © 2001 IFAC

Keywords: agents, cooperation, documentation, memory applications, socio-technical system design.

1. THE MULTI-AGENT COGNITIVE ENGINEERING FRAMEWORK

This paper presents an agent-based approach to organizational memory systems (OMSs). An OMS is composed of agents that now can be humans and machines. The agent-orientation of human-machine interaction is not new. Autopilots have been commonly used since the 1930's. Such agents perform tasks that human pilots usually perform, such as following a flight track or maintaining an altitude. Transferring such tasks to the machine modifies the original task of the human operator. Thus, the job of the human operator evolves from a manipulation task (usually involving sensory-motoric skills) to a supervisory control task (involving cognitive processing and situation awareness skills) (Sheridan, 1992). Software agent technology has made emerge a new type of interaction involving mediation, delegation, cooperation and coordination. Software agents enable users to center their interactions at the content level (semantics) partially removing syntactic difficulties. They also enable users to index (contextualize) content to specific situations that they understand better (pragmatics) and retrieve such content appropriately.
Various organization models can be proposed for a society of agents. The hierarchical organization often comes to mind because it reflects the way conventional organizations work in our occidential social world. Each agent has a specific task to perform. Usually an agent in the hierarchy assigns this task. At the bottom of the hierarchical structure, basic agents perform elementary tasks. The more agents near the top of the organization, the more their jobs are cognitive. A job is more cognitive than another when it involves more information and knowledge construction. Conversely, the more agents stay at the bottom of the hierarchical structure, the more they need to execute what they are told to do. Our industrial cultural heritage is strongly based on taylorism that is a very good example of hierarchical organizational philosophy. This philosophy is based on a top-down transfer of information and control that keeps working agents isolated from each other. Most agents are not globally responsible because they are not involved in the overall decision process. They are responsible for their job within the limited scope of their assignments.

An alternative to hierarchy is the jazz orchestra organization. In a jazz orchestra, each musician is an agent that performs an equally complex task. However, in contrast to the hierarchical organization where agents either control other agents and/or report to another agent, each orchestra agent acts according to a common set of rules (that can be transposed). These are coordination rules. These rules are defined by a syntax and a semantics. For instance, scores and music theory provide a set of such rules for musicians in an orchestra. In addition, there are transversal communication channels, and a single conductor manages the overall coordination. Musicians learn to interact and communicate with each other. Shared knowledge and skills are often very useful to individual agents for the benefit of the overall organization. All agents are globally responsible because they are involved in the overall harmony. They are responsible for their job as well as for the result of the group.

In the air traffic management domain for instance, the more pilots/aircraft become autonomous, the more they should become knowledgeable and informed agents. By autonomous, we mean that they know where they are located and what is their environment (situation assessment), and where they need to go according to the current and future situation (goal-orientation). In the past, the hierarchical organization of the airspace was necessary and sufficient because aircraft were not autonomous from a navigational point of view. Today (and tomorrow), this state is changing. Most aircraft are now equipped with global positioning systems (GPSs) that provide more autonomy (Stix, 1994). The GPS provides pilots with a more precise location of the aircraft, thus pilots’ situation awareness is improved in theory, and pilots are in a better position to decide by themselves (autonomy). This statement is valid if they trust the GPS. The orchestra concept is more compatible with this evolution than the hierarchical concept. Each aircraft is an agent and control centers are other agents that work as conductors.

My claim is that new technology brings more autonomy to aircraft if it also provides appropriate peripheral information to the pilots. This means that new cooperation and coordination rules need to be defined and incrementally refined. This observation holds for complex systems in general where situation awareness is a key issue. Organizational memory systems are living systems involving several kinds of agents. Human agents sometimes need to be aware of what other agents know. A crucial issue is focused on the existence and availability of those agents in the right context.

1.2 Various types of agent communication

Agent-to-agent interaction depends, in part, on knowledge that 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: (Class 2a) the agent knows its partner indirectly (using shared data for instance); (Class 2b) the agent knows its partner explicitly (using communication primitives clearly understood by the partner). This analysis leads to three classes of agent-to-agent relations:

• Class 1: non-cooperation;
• Class 2a: cooperation by sharing common data;
• Class 2b: cooperation by direct communication.

In the non-cooperation case, the agent is partially or totally ignorant of the behavior and reactions of the other agents. Conflicts may emerge when two agents compete for the same 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. The supervisor can be one of the partners or an external agent. Obviously, if available resources exceed the requirements of all the agents, conflict is automatically avoided.

In the case of cooperation by sharing common data, the agent knows that other agents exist and is aware of the results of (at least) some of the other-agents’ actions. Both of them use a shared database. Such a shared database can be an agent itself, i.e., a mediator, if it actively informs the various agents involved in the environment, or requests new information (self updating) from these agents. Agents use and update the state of this database. An example would be agents noting all their actions on a blackboard to which they all refer before acting. Agents have to cooperate to manage the shared database. This is no longer a problem of resource allocation, but a problem of sharing data which each agent can use when entitled to. This paradigm is called 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., shared data are generally supported by resources for which the corresponding agents may be competing.

In the previous cases, the interaction is always indirect. In the case of cooperating by direct communication, agents interact directly. They share a common goal and a common language expressed by messages, e.g., experts in the same domain cooperating to solve a problem. Agents communicate by incrementally constructing and sharing a common context. Each agent always attempts to construct a meaningful representation of the other agents in order to anticipate their behavior and reactions. Cooperation by direct communication involves learning about the other agents. Difficulty in or absence of learning may cause switching to either non-cooperation (and need a supervisor or a referee) or cooperation by sharing common data (and need a manageable interface understandable by each agent).

1.3 Flexibility and guidance in human/software-agent interaction

Flexibility and guidance in human/software-agent interaction are analyzed with respect to the three types of agent communication and the distinction between hierarchical organization and orchestra.

In the non-cooperation case, since agents know very little of the other agents, they need to be mentored by an OMS-knowledgeable agent. This type of agent can be an operation manual including organizational procedures implemented in the form of an intelligent assistant system.

In the case of cooperation by sharing common data, agents typically manipulate (interact with) clearly perceived interface objects that are familiar to them, e.g., menus, icons, electronic documents, and simulations. Interaction is more guided by the interface, and thus less flexible. A human agent who has less options to consider has less cognitive load, and does not need to be mentored by an OMS-knowledgeable agent.

In the case of cooperating by direct communication, agents interact directly with the other agents. They must share goals, representation of the situation and contexts. Active documents enable this type of interaction. The problem is no longer an issue of a static number of options available on the user interface. Software agents are designed to dynamically provide the right option at the right time using the right format clearly perceivable and understandable by users. Guidance becomes an intrinsic asset of software agents. The flexibility needs to be analyzed at the level of the organization of human and machine agents.

2. CONTEXT REPRESENTATION

2.1 The three levels of investigation

Mantovani proposed a model of social contexts describing the agent-environment interaction that is represented by the three following levels (Mantovani, 1996):

- Level 1: construction of the context, i.e., social context;
- Level 2: interpretation of the situation, i.e., everyday situations;
- Level 3: local interaction with the environment, i.e., artifact use.

Level 1 is more general than Level 2 which is more general than Level 3. At level 1, the social context is determined by values that are very general goals determined by the culture (Thomas & Alaphilippe,

---

1 In this paper, an artifact denotes anything designed by people.
1993). At Level 2, the psychological level, motivations are more precise but less persistent than attitudes, and are influenced by values and needs. At Level 3, the biological level, needs are biological strengths such as eating, drinking or sleeping. Mantovani describes Level 3 using the user-task-artifact triangle (Boy, 1998). He proposes equivalent triangles at Level 2, i.e., interests-goals-opportunities, and Level 1, i.e., action-history-structure. Mantovani’s three level model can be interpreted as follows:

- a user may have situated interests for action in a given context of action;
- the task that a user performs is based on situated goals coming from a social history; and
- the tool is built from opportunities that arise from a cultural structure.

### 2.2 Example of contexts in a design memory: legal issues versus design expertise reuse

In current human-centered design, we tend to store design history to provide explanations to other people on how an artifact has been designed and why design choices were made at the time they were made. Traceability of design rationale has become a crucial issue in modern industrial organizations for two different types of issues: legal and design expertise. Legal issues are related to the process of certification of an engineered system as well as to the process of incident and accident investigation. In a certification process, investigators have to show that the engineered system is safe, comfortable and efficient to use, for instance. Accident investigators need to find the reasons why a catastrophe happened. Designers are required to provide explanations that guarantee them legally from being prosecuted. The level of explanations is usually safety-related. Design expertise issues are very different. They involve finer grain of knowledge that people tend to hide. It is related to specific know-how that is often very difficult to formalize. Domain experts tend to be kept in their organization because this type of knowledge is unique and difficult to transfer. For instance, the number of remaining engineers who designed the Concorde supersonic aircraft has decreased to the point that the living design expertise on this domain has almost vanished. Since this type of knowledge is very precious, retired engineers are often solicited by the manufacturer to help new designers (Israel, 1995, 1998).

### 2.3 Spatial versus temporal perspectives

Organizational memory systems can be seen from two perspectives: spatial and temporal. Knowledge is usually spatially distributed within an organization. Cooperation/coordination or hierarchical mechanisms handle knowledge distribution within an organization for instance. Either people discover such mechanisms by themselves or they use an already existing communication mechanism provided by the organization. In hierarchical organizations for instance, people tended to create parallel cooperation mechanisms that are not provided by the organization. Information technology provides means to enhance communication using email for instance.

Knowledge is produced to be reused. But knowledge producers do not know who will need this knowledge and for what type of reuse. This makes it difficult for the knowledge producer to package his or her production. This is clearly a usability issue where the context of knowledge production should be represented as clearly as possible. For example, someone who needs to reuse design knowledge produced by Concorde’s designers more than 35 years ago may face a difficult problem of interpretation. He or she may call the person directly to obtain the context in which this knowledge was produced. This shows that context is an important issue in organizational memory systems. In particular, indexing should be context-sensitive.

### 2.4 Contextualization of information to memorize

People usually find it difficult to elicit and formulate what they know. In addition, they only express what they can according to their role within the organization, i.e., trust, power and hierarchical relationships strongly influence the elicitation process. Finally, it is not because we can formalize knowledge that it becomes available to the community. Additional work is always necessary to make the produced elicited information understandable and transferable. In particular, facts to be stored need to be contextualized (Israel, 1998).

This paper advocates the use of human factors dimensions for the contextualization of organizational memory systems. These dimensions can be included in one or several of the following classes of factors:

- Socio-cognitive factors such as novelty, delegation, cooperation, coordination, training, personality, qualification, new job design, roles
and work organization, efficiency, motivation, skills, control, ease of learning, information retention, complexity, reliability, errors and satisfaction.

- Physiological factors such as stress, headaches, muscular and biomechanical problems.
- Ergonomic factors linked to the interface such as input devices, displays, use of color, direct manipulation, graphics, natural language, 3D representation, operational documentation and multimedia.
- Environmental factors such as noise, air conditioning, lighting or ventilation.
- Productivity factors such as operator assistance, quality assurance, cost decrease, human error decrease, workload decrease, production time decrease and innovation increase.
- Constraints such as costs, time scales, budgets, personnel, equipment and building structure.
- Product functionality such as hardware, software and applications.

2.5 Contextual indexing

Software agent designers seldom know how people will use an organizational memory system. For instance, let us assume that you need some very specific information on the air conditioning in your house. The first thing you may try is to use the keyword "air conditioning" to find the right agent that would help to solve your problem. If you can specify the context of your problem, e.g., "you are a designer and are concerned by the connection of the air conditioning system, and have very little information about the electrical circuitry in your house", then you will probably find a list of vendors of pieces of equipment that may solve your problem. The solution will not be the same if you mention that "you are in your house and are freezing". You need to find a way to repair the air conditioning system. In other words, stating a problem requires good contextual conditions, if one wants to solve this problem easily and appropriately.

It is very difficult to elicit such contextual conditions since they are implicitly intertwined with the information being produced. They are often considered as common sense and thus do not qualify for explicit rationalization. However, if we consider the reasonable assumption that contextual knowledge is acquired incrementally, then incremental knowledge acquisition techniques are useful for on-line elicitation of context (Boy, 1991). Indeed, it is difficult for expert users to attach the right situation to any information retrieval strategy, simply because they do not remember well what they would do in a given situation. It is however, very easy to ask them to describe the relevance of retrieved information just after the fact (i.e., on-line elicitation). Obviously, the question is how to ask for such additional information from users without overloading or "annoying" them. One partial answer is certainly to reduce the amount of interaction users will have to perform to accomplish this additional task, or to provide specific high-quality interaction.

2.6 Active documents

The concept of active document is defined to support OMS users to easily rationalize, store, retrieve and understand any kind of information that is necessary at any time. The CID project is an example of integration of software agents into active documents (Boy, 1991). Direct manipulation improves the design and use of active documents. They facilitate OMS integration by satisfying conditions such as consistency of knowledge, internal consistency of the system that insures human reliability, context-sensitivity to the task, and expert advice when it is needed. Current documents are constructed from a variety of knowledge sources. They may have various formats according to the target and the available technology. The form and content of a document are both task-dependent (context of use) and domain-dependent (content). One of the main difficulties in designing active documents is to anticipate a very large number of contexts of use. Context of use is usually related to other entities such as situation, behavior, viewpoint, relationships among agents, discourse, dialogue, etc. Contextualization is extremely difficult using the conventional paper technology. It is made easier using computer technology when appropriate software agents are available or easy to construct.

Complementary documents are commonly used to understand original documents. If active documents are understood by the user without external help, then they are self-explanatory. In active documents, explanations should be formalized and transferred into a software agent that will help the user to better understand their content. For instance, in physics lab exercises, diagrams are presented to the students with missing parts that the students need to add in order to complete a consistent electrical circuit. On paper, these diagrams are presented to the student with a text
3. ACTIVE DESIGN DOCUMENTS

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 designed 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.

3.1 A concurrent approach to design and documentation

I claim 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 several persons must review papers that we write before being delivered outside. We also know that several persons must test artifacts 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 the design of interactive software. Writing words, phrases, paragraphs and chapters has become designing objects and software agents (Bradshaw, 1997). 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), 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.

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) (Boy, 1997), 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.

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 is a step towards the formalization of cognitive functions involved in the use of the artifact.

Cognitive function elicitation enables the design of interface objects (Boy & L’Ebraly, 1994; Broigniez, 1996; Boy, 1998). Cognitive function analysis enables the rationalization of the What, How, Why and How much of an artifact. The What provides the description of the artifact. The How provides the way one should use this artifact. The Why provides the design rationale. The How much provides an evaluation of the usefulness and usability of the artifact.

The active design document approach supports the process of designing concurrently an artificial agent and its operational procedures from the early stages of the design process (Boy, 1998). An active design document includes four aspects:

- 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.

3.2 Participatory design and traceability

The use of active design documents (ADDs) is an incremental process that leads to both participatory design and traceability. ADDs mediate interaction among design team members both in space and time. They are permanent records supporting communication between the actors involved in the life cycle of an artifact. Active design document’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 (Muller, 1991):

• 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; and
• 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 active design documents. The first approach is based on observation, the second one is based on cooperation.

By providing users with design-aid tools such as active design documents, 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 active design documents can be used in conjunction with methods such as the Group Elicitation Method (GEM) (Boy, 1997). Active design documents are continually modified with respect to opinions of various artifact life-cycle actors, evaluation criteria and domain requirements. When the active design document 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 (Carroll, Alpert, Karat, Van Deusen & Rosson, 1994). This approach reveals that indexing is a crucial issue to enable the traceability of design decisions that are included both in active design documents and in the relations between active design documents. The resulting library of active design documents 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 active design documents include design-centered interaction descriptions that document a preliminary task analysis, roughly sketched interface objects, and contextual links mainly defined by the design rationale based on a first set of overall requirements. Later in the life cycle of the artifact, interface objects become more sophisticated and user-friendly, interaction descriptions should become minimal, and contextual links richer in comments and feedback from tests. The easier the interaction with interface objects is, the shorter and crisper interaction descriptions are. An important issue involves how to handle the growth of contextual links. This is precisely where traceability problems arise. Contextual links should be classified, generalized and incrementally simplified (sometimes forgotten) in order to be used efficiently. A first solution is to group them by viewpoint.

4. DISCUSSION AND PERSPECTIVES

In this paper, organizational memory systems are a collection of people and artifacts. It takes into account
the emergence of computer-supported cooperative work and its repercussions on people within an organization. In addition, an OMS may vary according to the background of the design team and the requirements of the organization itself. We will start a discussion on a few possible views.

1. Research work is carried out in the knowledge management domain that includes requirements engineering, enterprise integration, use of artificial intelligence techniques for the development of OMS tools (e.g., knowledge acquisition, ontologies, data mining), and intelligent interfaces for knowledge retrieval.

2. Traceability of information within an organization will be discussed using the active document approach.

3. A list of OMS-related social issues of will be briefly presented.

4. The balance of the paper will present research perspectives.

4.1 Knowledge management and organizational memory systems

There are various approaches to knowledge management including the use of knowledge-based systems. In this paper, the objective is not to list the current systems that are developed or being designed because technology is changing so fast that when you will read this, these systems will probably be obsolete. Instead, the goal is to describe the key requirements, concepts, or challenges in developing an organizational memory system. An organizational memory system is not a static artifact that can be reduced to database management. It involves people and technology. An OMS evolves at all times, and facilitate information sharing and knowledge sharing.

For these reasons, four concepts were chosen as fundamental to describe it: agents, decisions, knowledge, and context. As already presented in the introduction, the concept of agent and cognitive function is crucial for the analysis, design and evaluation of an OMS. Cognitive functions are distributed among people, technology and organizational set-ups. These cognitive functions represent knowledge and know-how that the organization has produced. They shape the organization, i.e., the OMS as well as the organization itself are living entities characterized by their agents and the dynamic relationships among them that are incrementally constructed from the decisions agents make. Some agents may make minor decisions, others major ones. Agents exchange knowledge more or less formally according to the culture of the organization. Without context, this knowledge is useless. Since the difficulty is to contextualize knowledge, OMS research should focus on methods and tools that enhance the production of contextualized knowledge, as opposed to context-free knowledge. It is the price to pay to enable agents to trace knowledge within an organization.

OMS support tools enable people to manage formalized knowledge as well as semi-formal and completely informal representations that may vanish once they are used. Formalization is often obtained from classification and standardization of domain concepts and terms. When a concept is well accepted by a community of people, it can be used as a formalizing representation that will be shared by the members of this community. We cannot formalize everything. Only a few concepts can be elaborated and formalized. Among the rest, some pieces of knowledge can be represented using a semi-formal representation such as mark-up languages (SGML or HTML for instance), or ad-hoc structuring paradigms (hierarchical organization for instance.) Machine learning should play an important role in the future of organizational memory systems. Software agent technology is likely to help in this direction. Classification mechanisms can be extremely useful to categorize information.

Motivation to store and share information and knowledge within an organization is not guaranteed. Individuals may not see the real benefit of an organizational memory system. There should be a willingness to share knowledge. Human resource mechanisms should be discussed to improve cooperation and coordination. For this very specific reason, an ethnographic investigation of the culture of the organization is crucial. There are normative OMS functions that can be easily identified, but there are also meta-cognitive OMS functions that people use to work easier or faster, or for any other reasons that are related to the deeper culture of the organization. These meta-cognitive functions are essential to identify in order to better understand the structure and functioning of the organization.

4.2 Traceability functions and issues

Who is responsible for this decision? This is the type of question that one asks within a large organization everyday. The decision process within an organization is usually complex and results from the integration of
several micro-decisions. Some of these micro-decisions are sometimes important to trace back in order to understand the rationale of a synthetic information. Traceability is not a simple concept to define. It involves at least the following dimensions and related functions:

- design rationale management, i.e., record design objectives and justification of a design decision, make the design process explicit;
- legal rationale management, i.e., demonstrate compliance to existing rules, or get convinced that things are done correctly;
- context description, i.e., reserve time and project startup to use past experience (organizational memory), model the situation (roles, persons, reflexive behavior, viewpoints, constraints), agree about assumptions, or express design process towards a reader in order to become explicit – working in more contexts improves your sensitivity of context; various viewpoints (information and context descriptions that meet the customers needs);
- functions related to anonymous versus responsible reporting, i.e., responsibility engaged in writing personal information, responsibility for a decision, accountability;
- implicit versus explicit knowledge management;
- alternatives and decision criteria management;
- information granularity management (how heavy the documentation is, bureaucracy versus efficiency);
- indexing (adaptive, cost, knowledge should be contextualized);
- tutoring other people;
- remembering versus reinventing (knowledge reuse versus knowledge design).

Traceability involves issues such as:

- What do we need to document? Could we provide guidelines for things that are important to document, and other things that are not important?
- When do we need to document? We can document things as they occur, or document syntheses only.
- How do we need to document? Using paper, electronic media, or people. Using narratives or formal representations. Using a distributed network memory system or a single database.
- Why do we need to document? For legal reasons or knowledge reuse.

Traceability is not only based on an indexing mechanism, but also on important functions such as remembering active knowledge, not in interpreting passive information. Active documents are good candidates for the management of active knowledge.

### 4.3 Social meta-functions and issues

The following social issues are crucial:

- who defines the structure, function and content of an OMS;
- an OMS and the overall organization influence each other;
- an OMS should be defined according to the needs and goals of the organization and the personnel in the organization;
- objects available in the OMS should be sharable;
- knowledge transferred within the OMS should be controlled and evaluated;
- an OMS tends to facilitate transversal exchanges within an organization (is the organization culture ready to accept and manage them?);
- an OMS should be able to capitalize projects events across several work groups;
- individuals should be able to see the real benefit of an OMS.

### 4.4 Research perspectives

In this paper, we assumed that if technology is not a panacea for the organization, it can serve the proximal cause for mobilizing folk to actions (Soloway, 1995). Three main concepts have emerged: active documents, software agents and organization. Active documents are used as repositories of organizational and technical knowledge. People should be able to easily create active documents, as well as modify old ones. To facilitate active document design and publishing, libraries of software agents need to be created and maintained. Software agents are observers, information processors, and proposers. They can be active entities added to conventional documents transcribed into an electronic form. Some of them observe user's interactions, as it is the case in CID. They are able to produce actions from the data they

---

2A major issue is the interoperability of software developed in a specific software environment. Software agents should be platform-independent. Furthermore, the combination of object-oriented techniques (a software agent is a software object) and component-based software has some essential benefits listed by Rappaport (1995): reuse, extendibility, customization, distributability, and standardization. An example of standardization of agent-based software is given in (General Magic, 1994).
have acquired from the user. The action performed by
a software agent ranges from the activation of other
agents to the execution of (computational) operations.
Software agents are easy to manipulate and relate to
each other. They provide vivid behavior for a user
interface. They can be visible (audible), or invisible
(inaudible). When they are sensorial they have a
presentation shape (usually called a metaphor) on the
screen, or a sound, and a behavior. Otherwise, the user
does not know that they exist. In the field of
electronic documentation, agent adaptivity has been
shown to be extremely useful in information retrieval
(Boy, 1991b). In this case, software agents are
knowledge-based mechanisms that enable the
management of active documents. By manipulating
active documents, it is anticipated that the education
organization will evolve. It will produce a shareable
memory that can be capitalized by the corpus of the
organization.

In the aeronautical domain, Airbus Training has
implemented a procedure used by instructors that
enables them to provide experience feedback, i.e.,
instructors ask for improvements or corrections of
flaws in training tools based on the experience they
have on these tools. Experience feedback is based on
positive or negative experience that is interpreted by
training specialists to generate or modify corporate
knowledge. A corporate memory of the description of
the various pedagogical tools is maintained using this
procedure. The main point of such an organization is
the optimization of the end product destined for the
students.

Three theoretical issues should be better investigated:
• Information Technology (IT) shapes new
distribution of labor, i.e., whenever possible, a
cognitive function analysis of the organization
should be carried out to improve the allocation of
human and software functions – social repercussions of enterprise automation are not
easily controllable.
• Context representation, i.e., people productions
should be shared within the organization; the way
to reach this goal is to implement a situated
cognition approach instead of a pure goal-driven
(top-down) approach. Context representation is
useful to better handle information structuring,
indexing and retrieval.
• Memory models, i.e., a classification of use of
static information stores should be performed in
order to design them for maximum usability. A
careful use of human memory metaphors is likely
to help in the definition of an OMS. Knowledge
management tools and decision aids should be
based on memory models that users understand.
This is to improve the affordance of an OMS.

Three human factors issues need to be considered:
• Usability of organizational memory systems, i.e.,
OMS-specific usability methods and techniques
should be developed to test attributes such
deligation, control, trust, information
management, forgetting, coordination and
cooperation.
• Cooperation and coordination, i.e., better
understand if the processes of cooperation and
coordination need to be adapted to the culture of
the organization, and find out new resource
management paradigms. These tests should be
centered on an acceptable level of activity
(vigilance/stress) – sometimes cooperation and
coordination requires additional activities that may
not be compatible with the organization culture.
• Traceability of design decisions, i.e., improve the
way individual and group decisions are modeled.
Experts should be better modeled in order to take
them into account when recording information. A
distinction should be made between legal
knowledge and design expertise. Both should be
remembered but for different reasons. Usually, the
former is public, the latter is private.

Three technological issues should be better
emphasized:
• Human-centered design, i.e., technology should
not be developed without a cognitive function
analysis (agent modeling) that shows the pros and
cons of various function allocations. Rapid
prototyping and large-size testing are
recommended.
• Integrated technical documentation, i.e., as in all
user interface design, the procedure/interface
duality should be analyzed (Boy, 1998). The active
document approach is recommended both to
master the technology being developed and to
involve organization members into the design of
the OMS.
• Software agent technology, i.e., the evolution of
information management through computer
networks has promoted three important concepts
that are intelligent assistance (performance
support), adaptivity to users, and interoperability.
By definition, software agents incorporate these
three concepts.
ACKNOWLEDGMENTS

I thank my colleagues Mike Atwood, Jeff Bradshaw, Jonathan Grudin, Norbert Streitz, Bea Zimmerman, and the participants to the Industrial Summer School in Human-Centered Design of Organizational Memory Systems, held in August 1997 in the Southwest of France. They provided me with many comments and suggestions on the content of this chapter that was under construction at that time. I also thank my co-workers at EURISCO Alfred Attipoe, Markus Durstewitz, Martin Hellander, Rachel Israel and David Novick, as well as my Aerospatiale colleagues Brigitte Daniel, Jean-Pierre Heckmann and René Zimmermann for insightful discussions on the topic. Special thanks to Rachel. Many thanks to Helen Wilson for many useful and vivid exchanges on the topic.

REFERENCES


