Adaptive Mixed-Initiative Problem-Solving Assistants to support Collaborative Work in Flexible Organizations

Sylvie Delprat

System, Simulation and Information Technologies Department. Engineering, Learning Systems Team.
EADS CCR (Corporate Research Center), Toulouse, France.
sylvie.delprat@eads.net

Abstract

Different mixed-initiative problem-solving assistants have been implemented at EADS, but the requirements made on this type of application have evolved with the surrounding environment (collaborative work, distributed and evolving organizations). The focus of research work on this topic is now less on the sharing/control of the initiative and reasoning process and more on intelligent user integration/support and the adaptability of the assistants. This implies further research studies on context identification, intention recognition, adaptive and task-oriented user interfaces, ontology and knowledge modeling, with related information on context and task.

1 Introduction

Some years ago, several mixed-initiative problem-solving applications were developed at EADS for use in various domains, such as aircraft part and tool design (Delprat, 03) and cost estimation. However, the work environment, organization and processes have evolved, leading to new requirements being added and a new design of mixed-initiative assistants being envisioned. These changes and their related implications for assistants are detailed in section 2. Section 3 then presents a use case (with a description of the domain, the scenario and the processes), which is used to illustrate this new vision for assistant agents. The scenario deals with specification of simulation needs to validate aircraft system design and with the specification and design of simulators. Sections 4 and 5 then present the current thinking to provide a complete framework in order to assist designers and engineers in the most relevant way possible, without being too intrusive or tiresome.

2 Former approaches, evolutions and orientation/vision for future MI assistants

Former approaches. Past and current mixed-initiative assistants are dedicated to certain specific jobs and users. In former approaches, the process and task the job holder is to accomplish are defined once and for all for the job. The job holder knows the process, the input providers and the output consumers, and the available resources: information resources, documentation and the software tools dedicated to the job (see the first part of Figure 1).

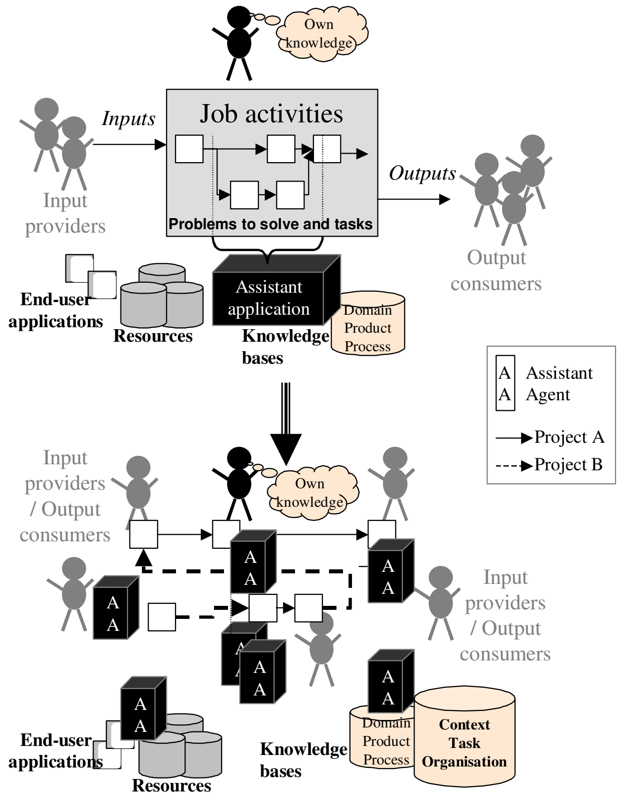


Figure 1. Typical processes in past and current organizations

A single application was built to support the user in all the sub-tasks of his job. The main aim of the assistant is to support the user on how to do and how to make the right decisions, by providing him with relevant knowledge and knowledge representative of the high-level expertise existing in the company. The main objectives and requirements were to:

- Guide and assist users throughout the job-related process

- Embed, in a kind of workbench, all useful functions for the job, a graphical user interface adapted to the job holders, and links or integration with external resources (software, product data management systems, catalogues….)
- Be adaptive to the level of user experience (from novice to expert), this implies providing a flexible user interface (from a complete guided process to answers on specific points) and to give the control to the actor on what it does best
- Provide and share knowledge between everyone performing the task, benefit from lessons learnt, and provide the fullest and most relevant knowledge possible. This implies interleaving different kinds of reasoning (human-, rule-, case-based reasoning or other AI-based reasoning).

The last two points led to the use of mixed-initiative reasoning systems, so that users may make their decisions/choices when they are experienced and they may let the system suggest choices/solutions to benefit from expert knowledge. Each actor can contribute to the task they do best (Allen, 1999).

On-going evolutions in organizations and overall processes. To remain at the leading edge of technology in its industrial sector, EADS has to work in cooperation with other companies and favor innovation. Thus EADS set up dynamic and highly flexible organizations and overall business processes. Carrying out projects/programs in cooperation with others induces to have distributed organizations and collaborative work, split across the partners and the supply chain (see part 2 of Figure 1). The project/program manager has to adapt the organization and the processes specifically for the current project/program according to the partnership and supply chain. Thus a job holder may sometimes have different responsibilities and changing tasks (of slightly different scope) for a specific project/program. To perform certain activities, a task A is performed by an organization/company X, and a task B is given to an organization/company Y for one project. Then, for a subsequent project, work sharing may be different and so the responsibilities of X and Y staff will not be the same as regards the activity.

Staff turn-over on some jobs implies that staff are not necessarily fully aware of the ability and behavior of all the functions of all the potentially useful software. Information resources, input providers and output consumers are specific to a project/program so, when assigned to certain tasks, they need to be made aware of the context and surrounding organization. They have a global idea of the task to be accomplished and the expected results, but they are not always aware of the different elementary problems to be solved for the task, where they can find input and information and who can provide them. In addition, they may be required to use software of which they have little experience. Hence, in such situation, the main aims of the assistants are to support users in finding right and required information, inputs and resources and to provide knowledge and assistance functions that are applicable and useful with regard to the current context and the problem to be solved.

The picture is now not so much a complex application that assists a user throughout his job task, but rather a set of wizards that help the user at various points in some of his currently allocated tasks.

This current context therefore implies that some additional demands are to be made on the problem-solving and mixed-imitative assistants, they need to:

- Be dynamic, highly modular and learn easily. Assistants must survive organizational and personnel changes.
- Be adaptive to different users and their specific role in the current process, contexts and tasks (problems to be solved).

The first point also implies that the development of mixed-initiative problem-solving assistants should be "light" in terms of cost and time. It is of no use developing a complex application holding almost all existing knowledge on topics and providing intelligent assistance for all tasks, because, when the application is completed, it will be useless as the organization and needs will have changed. So our interest is now to be able to develop these assistants in a short time, even if they are thus not as intelligent as they could be.

The second point means that complementary modules are required before being able to run mixed-initiative reasoning modules: context identification and recognition, intention recognition. Moreover knowledge and problem-solving task modeling should take into account and memorize information on the related context and tasks.

3 Illustration based on an aircraft system simulation scenario

System simulation is the process whereby aircraft systems are checked against their functional requirements and expected performance. Simulations may be performed with highly approximate simulation models, as well as with real equipment; this depends on the maturity of aircraft system development. The main actors (see Figure 2) in this process are:

- The *system designers* who have to design the system architecture and the systems and validate the system design against the requirements
- The *simulation engineers* who have to design and implement the simulator based on the simulation needs of system designers.

These main activities include different kinds of tasks: design, configuration, specification, checking. Some of these tasks are supported or should be supported by intelligent assistants. The role of these intelligent assistants is to provide the workers with the right information and knowledge and reasoning abilities adapted to user profile

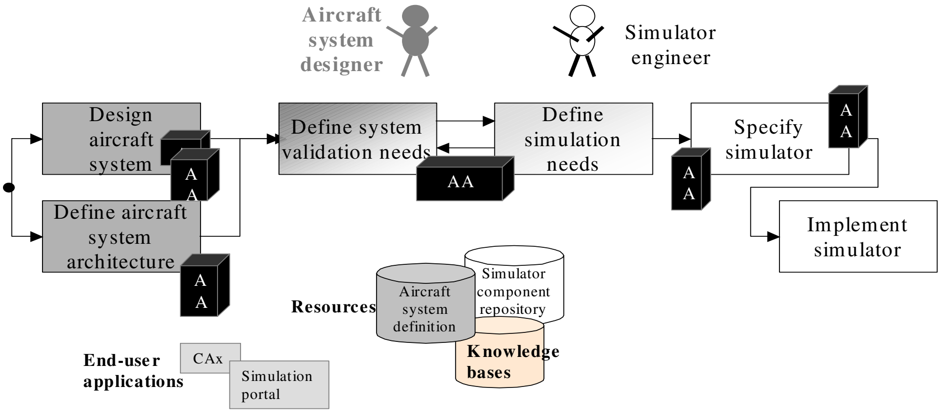


Figure 2. Presentation of the overall system simulation scenario

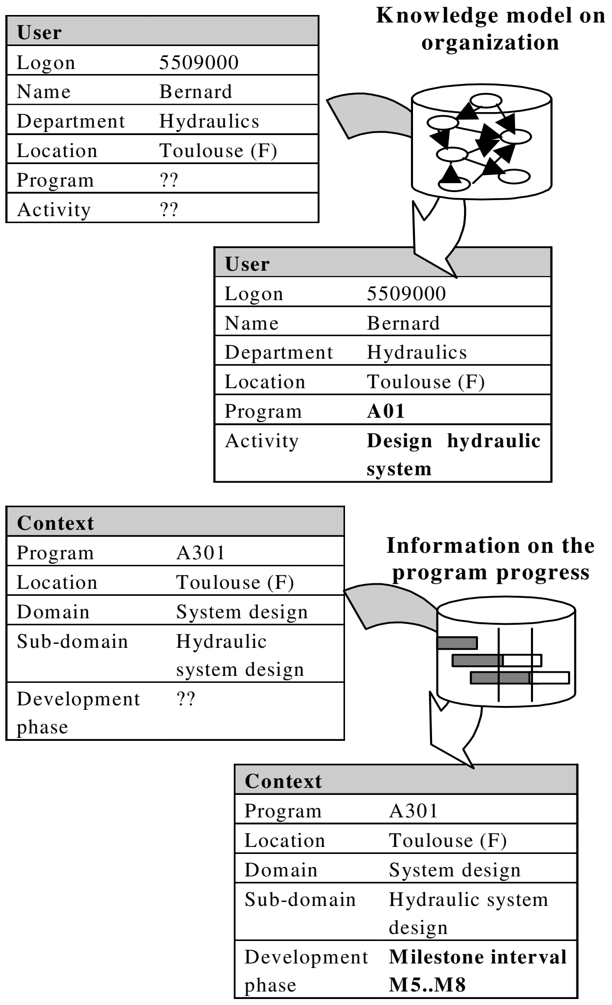
and current user assistance needs (i.e. adapted to the current context and the task to be solved).

4 Context and intention recognition

The objective is to be able to provide users with the mixed-initiative problem-solving assistants (and related applicable knowledge) relevant to the context and potentially useful for their current tasks. To achieve this, as described in section 2, some new complementary modules are now required: context identification, intention recognition, identification of problem solving tasks associated to needs for assistance.

4.1 User and context modeling. The idea is to automatically capture as much information as possible from user identification, computer data, and user actions in software. The "Intent Recognition assistant" (IR assistant) consolidates the information listed above and completes some objects aimed at describing the user and the context. Using this directly available information as a basis, the assistant completes the description by generating new information, either by inference or by querying the user.

4.2 Intention/problem modeling and recognition. Then the IR assistant tries to synthesize known information on user intent. It compares the information collected to characterize the current task and the known tasks described in the task base. It suggests a list of matching tasks or the closest tasks of possible relevance if no exact matching task is found (see the simple example in Figure 4). As in Coware (Introne and Alterman, 2003), the IR assistant provides users with a list of possible intentions, i.e. in our case, the possible tasks that the user wants to solve. When the user selects a task, the IR assistant will suggest a list of information resources and mixed-initiative problem solving agents that should be helpful for his task. The user selects the main assistant agent that will best support him throughout the task to be performed. This main assistant


Knowledge model on organization
Information on the program progress

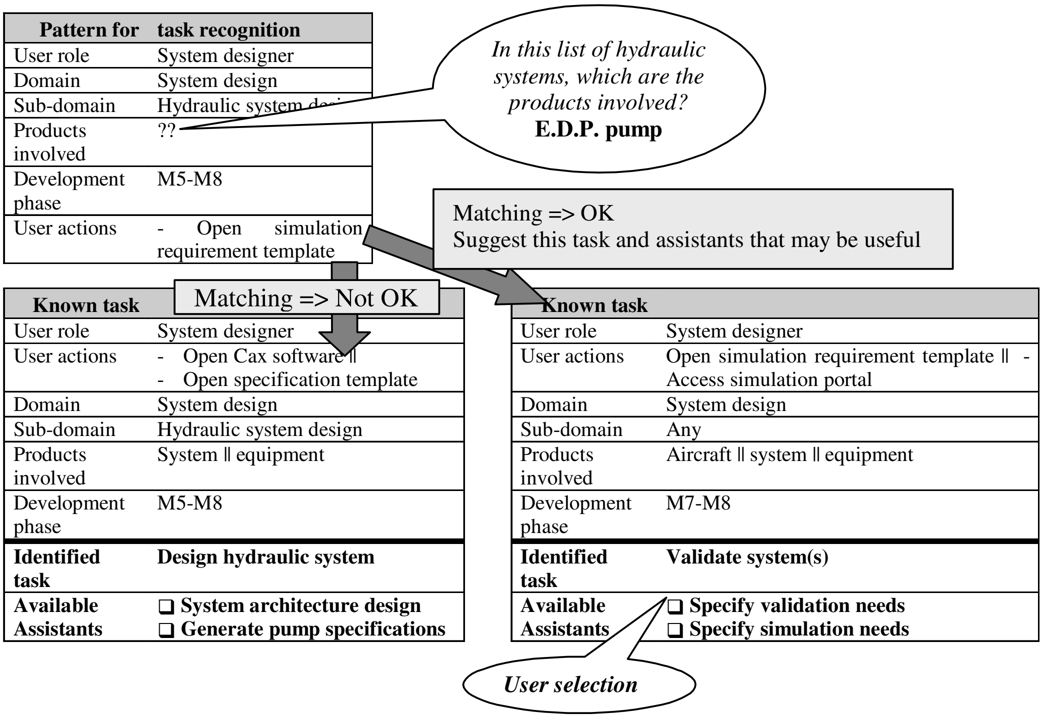
User	
Logon	5509000
Name	Bernard
Department	Hydraulics
Location	Toulouse (F)
Program	??
Activity	??

User	
Logon	5509000
Name	Bernard
Department	Hydraulics
Location	Toulouse (F)
Program	**A01**
Activity	**Design hydraulic system**

Context	
Program	A301
Location	Toulouse (F)
Domain	System design
Sub-domain	Hydraulic system design
Development phase	??

Context	
Program	A301
Location	Toulouse (F)
Domain	System design
Sub-domain	Hydraulic system design
Development phase	**Milestone interval M5..M8**

Figure 3. Completion of user and context models



Pattern for	task recognition
User role	System designer
Domain	System design
Sub-domain	Hydraulic system desi
Products involved	??
Development phase	M5-M8
User actions	- Open simulation requirement template

Known task	
User role	System designer
User actions	- Open Cax software || - Open specification template
Domain	System design
Sub-domain	Hydraulic system design
Products involved	System || equipment
Development phase	M5-M8
Identified task	**Design hydraulic system**
Available Assistants	**❑ System architecture design** **❑ Generate pump specifications**

Known task	
User role	System designer
User actions	Open simulation requirement template || - Access simulation portal
Domain	System design
Sub-domain	Any
Products involved	Aircraft || system || equipment
Development phase	M7-M8
Identified task	**Validate system(s)**
Available Assistants	**❑ Specify validation needs** **❑ Specify simulation needs**

In this list of hydraulic systems, which are the products involved? **E.D.P. pump**
Matching => OK
Suggest this task and assistants that may be useful
Matching => Not OK
User selection

Figure 4. Matching information on the current task and known tasks

agent often calls secondary agents in interleaved manner to support some specific elementary task (see the illustration on the system simulation scenario in Figure 4).

To avoid the assistance becoming a nuisance, the reasoning mechanism to identify the possible user intent or the potentially useful assistants should be enhanced and should be able to learn. This may be done by adding probabilistic information (for example by Bayesian Belief Networks, Jenson, 1996) or by learning from past task/assistant suggestion attempts (for example by case-based reasoning where a case will represent each use of the IR assistant, the user's answer and the task/assistant selected by the user).

4.3 Adaptive and task-oriented user interfaces. The core of the MI assistant agent should set up a process specific to the task to be solved, but independent of a specific user or context, since the MI agent may be used by different users with different roles.

Thus, when the IR assistant detects the user profile, the context and the task, it should also give the selected MI assistant agent the customization parameters to allow the agent to adapt itself to the current situation. The items that will change from one situation to another are, for example, user interface, knowledge to be applied, expected outputs. For example, the specification of a system simulation will not be as detailed for a simulation in the preliminary development phase as for a simulation just before manufacturing the first aircraft in the program. In the second case, for example, the operating/flight phases covered in the simulation should be extremely precise.

The MI assistant agent interface should be adapted to the task and to mixed-initiative query elaboration and problem-solving, but it should also be adaptive to the current context and user. For example, the system designer and the simulation engineer may use the same MI assistant for the specification of validation needs, but their expectations are not exactly the same. The simulation engineer will use the assistant to check the completeness and the consistency of the validation needs to ensure that the information on validation needs will enable the simulator to be designed and implemented. On the other hand, the system designer will use the assistant to attempt to formulate his validation needs for the simulation engineer but, above all, to guide him and confirm his validation choices according to the aircraft system requirements and to system design maturity.

5 Query elaboration and problem-solving process - Shared ontology

Problem-solving process. For different user profiles (for example, according to level of expertise), the agent should be designed to be used either by a user (with an expert profile) who wants to keep the main initiative (and where the agent enters some details, checks/revises user decisions and supports the user for limited choices) or by a user (with a novice profile or for routine tasks) that lets the agent drive the problem solving process and perform the task in almost fully automatic and autonomous manner (and asks the user specific questions to confirm an orientation or a decision).

In the query elaboration phase, query formulation should be shared between the agent and the user. For example, when the simulation engineer has to specify the simulator on the basis of validation and simulation needs, the simulator specification agent interleaves the reasoning of inference engine, case-based reasoning and user actions (choice -direct input from the user-, selection -among potential solutions suggested by the agent- or validation -of the agent decision-). To do this in a flexible manner, the MI agent has to implement the required properties of the proposed framework (Gupta and Aha, 2003) for incremental query formulation.

In the system simulation use case application, this approach (for the mixed-initiative problem solving process and query elaboration phase) is used for the "Specify simulator" task (described in Figure 5). In some previous assistant applications (Delprat, 2003; Duprieu et al, 2000), most of these properties were implemented, but the main change is that we are now working with fully collaborative and distributed processes, so some interleaved assistant agents

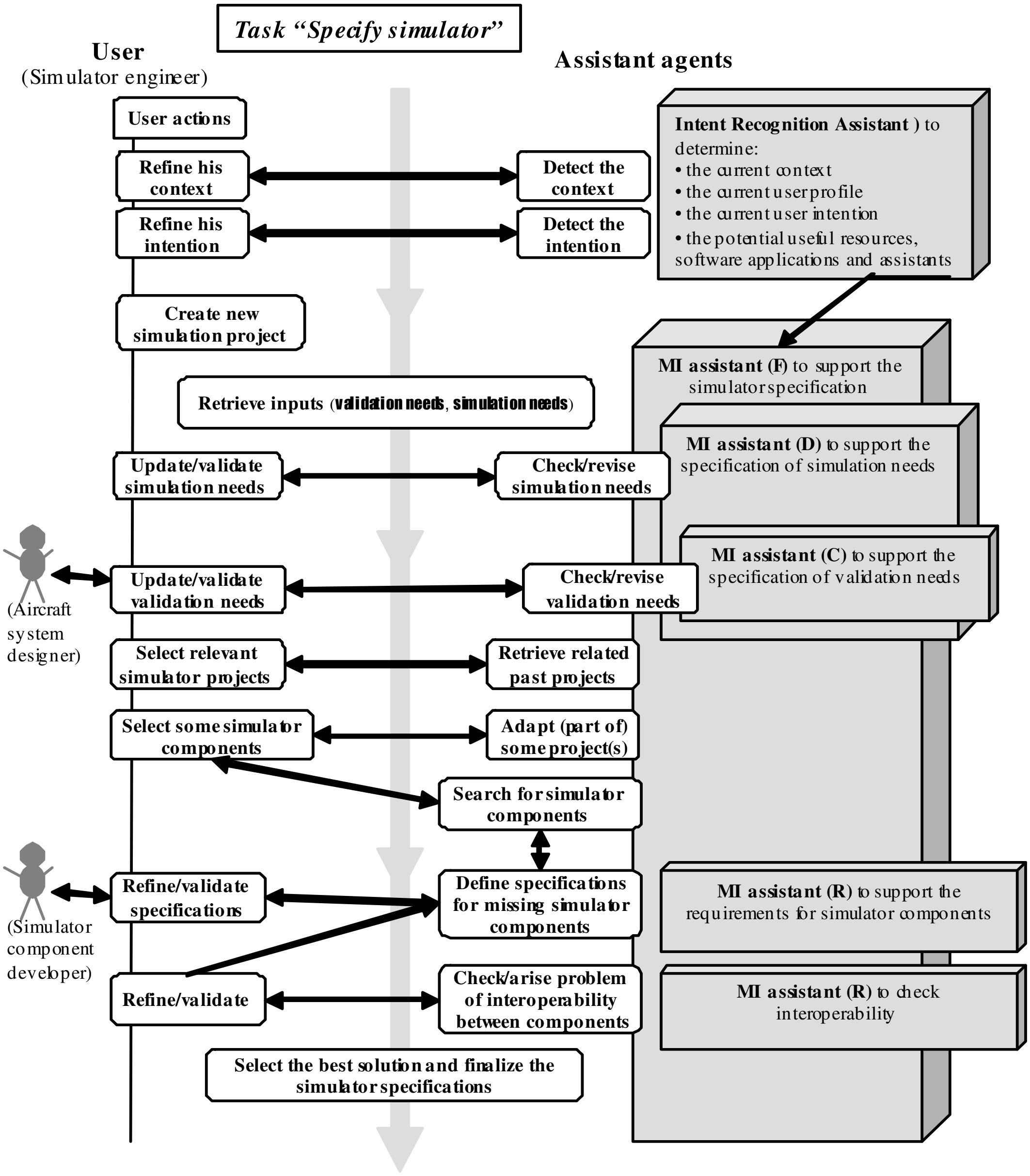

Task "Specify simulator"
User
(Simulator engineer)
Assistant agents
User actions
Refine his context
Detect the context
Refine his intention
Detect the intention
Intent Recognition Assistant) to determine:
• the current context
• the current user profile
• the current user intention
• the potential useful resources, software applications and assistants
Create new simulation project
MI assistant (F) to support the simulator specification
Retrieve inputs (validation needs, simulation needs)
MI assistant (D) to support the specification of simulation needs
Update/validate simulation needs
Check/revise simulation needs
MI assistant (C) to support the specification of validation needs
(Aircraft system designer)
Update/validate validation needs
Check/revise validation needs
Select relevant simulator projects
Retrieve related past projects
Select some simulator components
Adapt (part of) some project(s)
Search for simulator components
(Simulator component developer)
Refine/validate specifications
Define specifications for missing simulator components
MI assistant (R) to support the requirements for simulator components
Refine/validate
Check/arise problem of interoperability between components
MI assistant (R) to check interoperability
Select the best solution and finalize the simulator specifications

Figure 5. Mixed-Initiative Query Elaboration and Problem Solving process for the task "Specify simulator"

may be used in different situations. This implies that we need some generic frameworks and components to build the MI assistant agents, such as in the I-X research program (Tate, 2003), among these we find: the core architecture and module that will enable the development of dynamic, flexible and modular agents, the user interface building blocks to provide the user with a graphical interface adapted to the task, and the module to create and update the knowledge models. These knowledge models should be shared and used by different agents, thus resulting in some constraints on the underlying ontology.

Ontology for shared models between humans and mixed-initiative agents. This ontology should separate the different types of knowledge:

- Knowledge of the objects (domain items, products, processes) that will be manipulated by the agents and users during the task and problem solving process. This knowledge should be described independently of a specific task or use.
- Knowledge about the tasks, the contexts and the user profiles.
- Knowledge related to the problem-solving processes (constraints, issues to be solved, planning of the problem-solving steps, reasoning methods…).

Until now the Moka methodology (Moka consortium, 2001) has been used to model knowledge on domains, products and processes for knowledge-based design applications. We need to upgrade the Moka related ontology to meet the new requirements. <I-N-C-A> ontology (Tate, 2003) may be a basis for this requied improvement.

6 Conclusion and research issues

The main aims of assistants has changed from providing expert level knowledge and experience giving the level of expertise of activities for one job to dynamically assisting users in their different roles and elementary tasks. To cope with the evolutions in organizations and to support collaborative work, assistants must change from certain local applications to light intelligent assistant agents, integrated in flexible and dynamic manner and supporting the whole distributed process.

There are numerous related current research topics dealing with implementation of the vision presented here. However, many research issues still remain to be addressed, notably: context recognition, intention recognition, ontology and shared knowledge repositories with information of related applicability (to a context, a task…), adaptive and task-oriented user interfaces.

References

Allen, J. 1999. Mixed-Initiative Interaction. *IEEE Intelligent Systems*, 6, (pp.14-16).

Delprat, S. 2003. Mixed-Initiative Assistants for Aircraft Tool Design. In *Proceedings of the Mixed-Initiative Case-Based Reasoning Workshop held at the Fifth International Conference on Case-Based Reasoning*.

Duprieu, B., Delprat-Courbot, S., Caillaud, E., & Fortunier, R. (2000). A Knowledge-Based System for Design of Deep-drawing Tools. In *Proceedings of the French Conference on Knowledge Engineering*.

Gupta, K.M. & Aha, D.W. 2003. A Framework for Incremental Query Formulation in Mixed-Initiative Case-Based Reasoning. In *Proceedings of the Mixed-Initiative Case-Based Reasoning Workshop held at the Fifth International Conference on Case-Based Reasoning*.

Introne, J. and Alterman, R., 2003. Supporting Intent Awareness in Groupware. In *Proceedings of 18th International Joint Conference on Artificial Intelligence*.

Jensen, F. 1996. *An introduction to Bayesian Networks*. Springer.

Moka consortium 2001. *Managing Engineering Knowledge. MOKA: Methodology for Knowledge Based Engineering Applications. Professional Engineering Publishing.*

Tate, A. 2003. <I-N-C-A>: a Shared Model for Mixed-initiative Synthesis Tasks. In *Proceedings of 18th International Joint Conference on Artificial Intelligence.*