Mixed-Initiative Assistants for Aircraft Tool Design

Sylvie Delprat

Knowledge Engineering Department
EADS CCR (Corporate Research Center), Toulouse France
sylvie.delprat@airbus.com

Abstract. This paper presents an approach to building different interactive assistants for tool and equipment design in the aircraft industry. The scope is the upstream design process ranging from the conceptual design (general design principles) to the detailed specification of the tool or equipment. This approach interleaves three main actors (the user, a rule-based inference engine and a case-based reasoning system) and supports the sharing of initiative between these different actors. The aim is to highlight (based on lessons learnt from experiments on industrial cases) the benefits of such an approach, and also to point out the limitations, the needs for improvement and the key research issues to be solved so that it can be applied more easily in industrial applications.

1. Introduction

The design process is a complex task in which knowledge cannot be made completely explicit, and which relies mainly on designer experience. In the aircraft industry, time between different aircraft programs varies and can be long, leading to risks of forgetting and loss of expertise. Novices can learn really efficiently only if they can participate actively in the development of an aircraft program with some expert designers. So there is an important need for them to have support tools that capture expert knowledge and can assist then in an interactive and natural way.

Many Artificial Intelligence techniques have been used in design applications (e.g., Maher & Pu, 1997; Sycara & Navinchandra, 1992). The design activity interleaves several types of knowledge, including rules (physical principles, best practices, preparation of numerical calculation and simulation, interpretation of the results, queries into databases and component libraries), previous design cases and problems encountered after design. Cooperative reasoning techniques (rule- and case-based reasoning techniques) permit the representation of a wider variety of knowledge towards improving the quality of the system's results. However, whatever different complementary reasoning modes we add, the main actor of the design process should remain the designer, and the knowledge-based application should be considered as an "intelligent assistant", which would be aware of all the past experiences and the know-how of all corporate experts in the domain.

Furthermore, the knowledge-based application should be adaptive and useful for all the designers (i.e., both novices and experts). Obviously the use of the knowledge-based assistant and the related expectations will not be the same for both groups. The

novice should be guided step by step through the design process, expecting to have access to all the captured knowledge and to have solutions suggested. The experienced designer will use the assistant only on very specific tasks of the process or to corroborate his choices. So the user interaction with the assistant for tool design is a key issue to be solved to increase the likelihood of the system being accepted (Berry & Broadbent, 1987; Patil et al., 1982) and provide satisfactory results and return on investment. A mixed-initiative approach permits the system to be adapted to each user, ranging from users who require little support (but can be important knowledge suppliers) to those who need maximum guidance and support (McSherry, 2001; Aha & Gupta, 2002).

In Section 2, we describe the context of knowledge-based engineering applications in design. Then in the following sections, we present the features of the interaction between the user and the assistant and we conclude by the outcome of our mixed-initiative approach to the implementation of interactive assistants.

2. Knowledge-Based Reasoning Aided Design

The approach presented in this paper is based on our work on assisting designers in their tasks of design, specification and document authoring of aeronautical products and tools. The targeted knowledge-based engineering applications have to assist (based on knowledge capture and reuse) some design tasks while integrating multidisciplinary expertise domains (especially manufacturing, inspection, cost and customer support constraints early in the design phase). These applications process inputs such as part specifications, user choices, external constraints and context information to generate elements of solutions (e.g., detailed definition, CAD model, document, cost estimation, process planning) driven by a modelled process and user actions, and by reusing best practices and standard/repeat components. The approach presented here has been implemented and evaluated on two main use cases: in the manufacturing domain (design and specification of forming tools (SCORE system) (Duprieu et al., 2000) and in the aircraft maintenance domain (equipment design and generation of specification documents (GSESA system)). In these systems, case-based reasoning systems, inference engines and users share the initiative of action and suggestion, and progressively build a collaborative solution. As in other applications and decision support tasks (e.g. planning (Muñoz-Avila et al., 2000)), our systems integrate user interactions, generative techniques and case retrieval, since no single approach (pure generative, case retrieval) can support the whole problem-solving process.

2.1 Global Problem-Solving Process in Design

In order to reduce the complexity of design problems, the global process has been subdivided into elementary tasks. This allows us to mix more tightly and subtly the reasoning actions and to follow closer the designers' activities. These elementary tasks operate a (question-answering) dialogue related to the different steps of problem solving:

- problem description, analysis, and classification, and
- solution generation and selection.

Figure 1 locates the different reasoning techniques against each step of the problem-solving process.

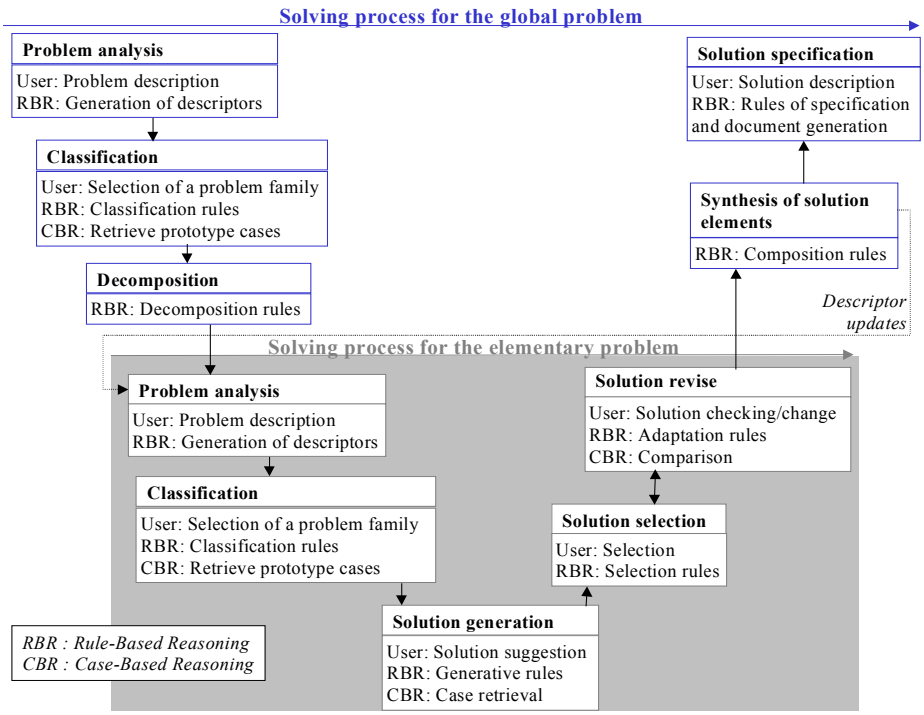

Fig. 1. General problem solving process

2.2 Classification of Knowledge in Design Applications

The main knowledge used in design can be classified as follows:

- *Cases*
  - design instance/cases: hold knowledge specific to each experience, design choices, solution description and rationale, encountered problems and cost,
  - libraries of standard/repeat components and knowledge to search the best fitting component,
  - design and configuration tables, which hold different configuration alternatives.
- *Rules*
  - general design rules, representative rules of the best practices,

- rules integrating multi-disciplinary expertise (manufacturing, customer support),
- general knowledge about physical principles, cost and technical limits,
- rules that prepare data and interpret the results of numeric calculation and simulation.

- *Checks and constraints*
- regulation constraints (e.g., European regulation about electrical machines for ground support equipment).

3. Different Modes of Interaction

Many mixed-initiative applications in Case-Based Reasoning rely on dialogues and a conversational approach. In our work on design, the dialogue does not rely on a grammar approach or textual conversation as in (Bridge, 2002). Our systems offer interfaces as close as possible to the ones familiar to the designers and the dialogue is operated via graphical components (e.g., toolbar, buttons and icons, lists, schematic drawings).

3.1 Dialogue Features

We use here, as a template, the list of dialogue features described by McSherry (McSherry, 2002) to present briefly some features of the interactions in the SCORE and GSESA systems.

- *Volunteering data*

The user can volunteer data when he wants, in the order he prefers, as detailed as he wants. The user has only to give data that cannot be extracted (by queries in database, in product data management systems or CAD model) nor inferred from data already given by the user (Aha et al., 2001; Aha & Gupta, 2002). This allows the user to answer only to the key questions (i.e., the most useful and relevant ones in his mind for the current problem-solving process step).

- *User in control*

The user can select the process step or be guided more automatically along the design process. The user can decide who will drive the solution elaboration:
- he can suggest solution elements and ask the system to check and to corroborate based on the past experience, or
- he can let the system suggest solution elements, which he can validate or modify.

- *Intelligent question selection*

To avoid asking many questions, a selection of questions (those which are the most relevant ones in most of the cases for the current step) is made a priori. Other questions may be less useful or important only in certain cases. These additional questions (named *supplemental questions*) are only asked to the users when it is really essential to discriminate the solutions and to suggest the most adequate ones.

- *Tolerating incomplete data*

While answering the questions, the user may decide to skip a question (no answer) or to say that he cannot answer (by an "unknown" value).

- *Updating data*

At any time, the user can revise and modify his previous choices. Then the system updates the inferred data and detects which questions should be asked again to the users. This option allows the user to evaluate different solution alternatives or to understand the links between the problem descriptors and the solution impacts.

3.2 Dialogue between the User and the System

Figure 2 shows the main sharing/swapping of control and interactions between the user and the assistant (SCORE or GSESA system).

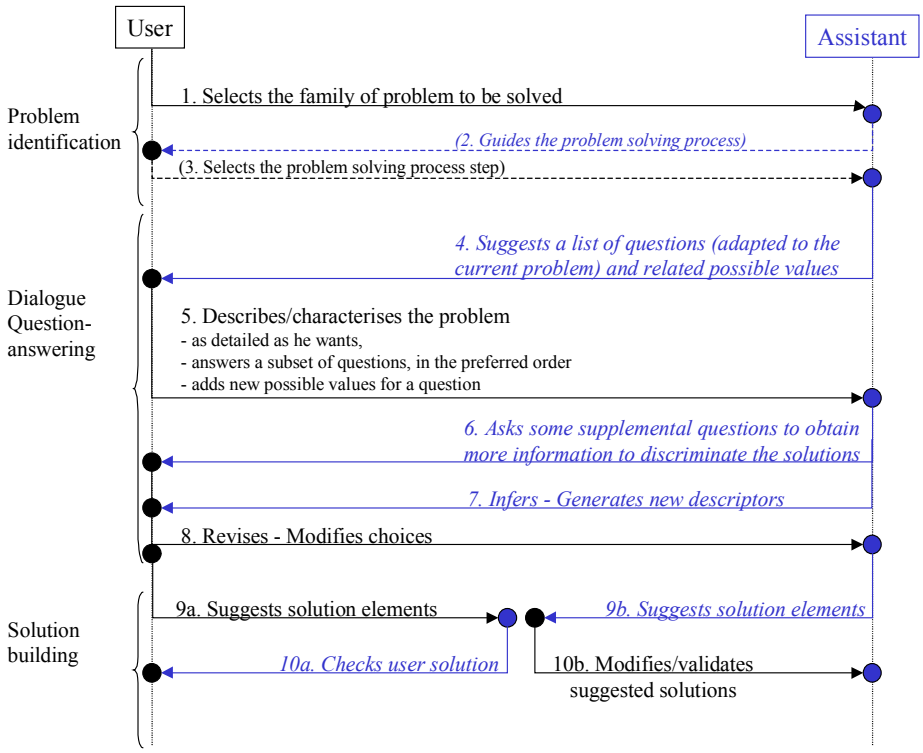

Fig. 2. Possible contributions of both the user and the assistant

Figure 2 presents the possible actions in a logical order (to progress towards the solution), but in fact there is no fixed and predefined sequence of actions. For example,

whenever the user wants, he can have solution suggested or enter a new piece of information. Obviously, if the user enters all the information before asking for a suggestion, the assistant will be more accurate. The different actions that could be performed by the users and by the system may be summarised as follows.

The *user* may:
- identify and characterise the domain and the type of problem to be solved (1),
- identify and select the process step in which he requires support (3),
- describe/characterise the problem (as detailed as he wants, in the order he prefers, for the subset of descriptors he wants) and add new possible values (knowledge updates) for a question (5),
- revise, modify his choices (8),
- suggest elements of solution (9a),
- modify or validate the solutions suggested by the system (10b).

The *system* may:
- guide the problem solving process (2),
- suggest a list of questions (adapted to the current problem) with related possible values (4),
- ask some supplemental questions to obtain more information to discriminate the solutions (6),
- infer, generate/deduce automatically new descriptors (7),
- suggest solution elements (9b),
- check/corroborate the solutions suggested by the user, display some examples of previous similar solutions (10a).

3.2 Mixed-Initiative Control During Solution Elaboration

The actions marked with a and b are related to two different modes, depending on which actor has the main initiative elaborating the solution. In the first one (mode *a*), the user takes the main initiative and suggests the solution, and the system supports the user in different ways:

- by retrieving some past design cases similar to the problem and to the solution given by the user,
- by applying some check rules to verify if the solution element is consistent with the data and respects the constraints,
- by corroborating the user choices. The user does not have to launch intentionally a CBR search. For each user's decision, the system launches a CBR search in a transparent way, and stores the results (the most similar cases and the similarity score). If the system detects after a new user's decision, that the similarity score of the most similar case has dramatically decreased, it will display a warning, explain the reasons for this warning and ask the user to confirm his decision and to give some justifications (that will be reused later).

In the second mode (mode *b*), the knowledge-based system takes the main initiative and suggests one or several solutions, which the user can select, validate or modify.

3.3 Co-operation and Collaborative Decision Making

Our aim is to make different knowledge-based reasoning actors (i.e., in our systems, these are rule engines tightly linked to numerical calculation, case-based reasoning systems, and users) really co-operate. By subdividing the problem-solving process at a very detailed level (cf. figure 1), where at each step, each knowledge-based reasoning actor can contribute to the task what it does best (Allen, 1999), this yields very interactive assistance, where the user is free to make his own decision and can get suggestions, benefiting from the past experience, which is illustrated with past examples and rationale and is linked to the general knowledge in the domain. So each actor can suggest and make the problem-solving process progress step by step. Currently the collaboration is effective between one user and the different knowledge-based reasoning modules. A possible improvement would be to enable the collaboration between different users and the knowledge-based system.

4. Knowledge Base Maintenance and Integration of User Feedback

The systems should reduce the needs for maintenance and must not require fastidious and time-consuming operations to feed the knowledge bases. Thus the knowledge-based system has to capture automatically as much information and knowledge as possible. So each user choice or action is analysed and related information is stored in the case base.

The SCORE and GSESA systems integrate easily the user feedback after the tool design (e.g., encountered problems during the tool finalisation, manufacturing, use or maintenance). The description and analysis for each problem are stored and linked to the design case, and if it is the case, linked to the design choice which caused the problem. The estimated and real costs are also stored and linked to the design case in order to provide a cost basis for a new design.

5. Conclusion and Main Research Issues

For both systems (SCORE and GSESA), prototypes have been developed and evaluated by designers in operational conditions (in manufacturing unit and in design office). This evaluation highlighted satisfactory results: these systems are well accepted by the different users (both experts and novices) and respond properly to the user needs.

We indeed observed different ways of using the assistants according to the experience of designers. For example, the SCORE system was used to support the training period of some new designers, to help them (step by step) to analyze the design problem and the forming strategies, and to accomplish the design process. At each step of the design process, the novices expect advice and suggestions of solution from the assistant. On the other hand, experts use the SCORE system only to check their ideas of solutions at specific design steps. They describe the different solution options and they search for previous design cases to corroborate, this permits to avoid some trials

of forming strategies, which led in previous tests to manufacturing problems. The first system is now operational, and the second one should be industrialized during this year.

The main benefits of our mixed-initiative approach to the implementation of interactive assistants in this domain are:

- to obtain higher quality results from the collaboration and contribution of each actor (user and knowledge-based reasoning modules) where it is the most competent,
- to be suitable for domains whose knowledge cannot be made completely explicit/formalised,
- to have 'natural" and flexible software adapted to each user profile, to different levels of user expertise, and consequently having well accepted user-centric systems (where the user feels that he remains in control),
- a system reactive to different types of needs (from the very specific and punctual analysis and advice to complete assistance and training in the design process).

However, some key research issues remain so that this kind of application can be used more easily in an industrial environment:

- *reduce the initial investment to set up knowledge bases that can be used by different actors*. One problem is to make the different knowledge reasoning actors communicate and exchange knowledge; there is no standard representation language for all the actors (rule- and case-based reasoning and human reasoning),
- *balance user freedom and maintenance needs*. In order to facilitate the maintenance of knowledge-based systems and limit the maintenance tasks, we try to capture knowledge in the most automatic and transparent way as possible. To achieve this, we have chosen to capture user choices and actions, based on user answers, so if the user skips some questions, the problem is how to access and store the missing information for reuse.
- *support more flexible and dynamic processes*. In our first systems, the design process was modelled and the dialogue possibilities were built on the basis of this process. For several years, we have worked to support more flexible and dynamic design processes.
- *improve the interaction with the CAD environment.* The objective is to extract information from CAD models to avoid duplicating inputs and to also account for user actions related to the geometry.
- *support collaborative decision making* and manage the sharing of initiative among several users.

References

Aha, D.W., Breslow, L.A., & Muñoz-Avila, H. (2001). Conversational Case-Based Reasoning *Applied Intelligence* (pp.9-32).

Aha, D.W., & Gupta, K.M. (2002). Causal Query Elaboration in Conversational Case-Based Reasoning. *Proceedings of the Fourteenth International Conference of the Florida Artificial Intelligence Research Society*. AAAI Press, Florida.

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

Berry, D.C., & Broadbent, D.E. (1987). Expert Systems and the Man-Machine Interface. Part Two: The User Interface. Expert Systems 4.

Bridge, D. (2002). Towards Conversational Recommender systems: a Dialogue Grammar Approach. *Proceedings of the Mixed-Initiative Case-Based Reasoning Workshop held at the Sixth European 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. *Proceedings of the French Conference on Knowledge Engineering*.

Maher, M.M, & Pu, P. (1997). Issues and applications of Case-Based Reasoning in Design. Lawrence Erlbaum Associates, NJ.

McSherry, D. (2002). Mixed-Initiative Dialogue in Case-Based Reasoning (2002). *Proceedings of the Mixed-Initiative Case-Based Reasoning Workshop held at the Sixth European Conference on Case-Based Reasoning*.

McSherry, D. (2001). Interactive Case-Based Reasoning in Sequential Diagnosis. *Applied Intelligence*.

Muñoz-Avila, H., Aha, D.W., Breslow, L.A., Nau, D. S., & Weber, R. (2000). Integrating Conversational Case Retrieval with Generative Planning. *Proceedings of the Fourth European Workshop on Case-Based Reasoning*.

Patil, R.S., Szolovits, P., & Schwartz, W.B. (1982). Modeling Knowledge of the Patient in Acid-Base and Electrolyte Disorders. In: Szolovits, P. (ed) *Artificial Intelligence in Medicine*. Westview Press, Boulder, Colorado (pp. 191-226).

Sycara, K. P. , & Navinchandra, D. (1992). Retrieval Strategies in a Case-Based Design System. *Artificial Intelligence in Engineering Design*, Vol. 2, (Eds) C. Tong & D. Sriram, Academic Press, (pp. 145-163).