A collaborative architecture for design evaluation and refinement

M. V. Belmonte, L. Mandow, J. L. Pérez de la Cruz, R. Conejo and R. Morales
Dpto. Lenguajes y Ciencias de la Computación
Universidad de Málaga
Plaza de Ejido s/n
Malaga 29013
Spain

1. INTRODUCTION

Since the birth of Expert Systems technology, multiagent systems have been studied and applied to solve real-world problems (Lesser & Herman 77). However, a flourishing of multiagent or collaborative systems can be detected during last years. Blackboards systems (Engelmore & Morgan 88) have been developed with different control architectures and in many application areas. On the other hand, the new paradigms of distributed AI (Bond & Gasser 88) and computer-supported cooperative work (CSCW 90) have broadened the field both by decoupling the knowledge sources or problem solvers and by allowing the intelligent interaction of human and non human problem solvers.

Actually there are some application domains where cooperative systems have been successfully used. Its application has been necessary in the resolution of some problems because of their clear collaborative nature. Some of this applications area are (Durfee et al. 89):

- Human cooperation aided by computer, which tries to provide a support in the activities where the people of an organization have to cooperate, such as help to make decisions, communications etc..One of the assistants of this kind is POLYMER (Croft & Lefkowitz 88). This system used an integrated representation of activities and some others objects in its environment, allowing a flexible execution of activities by means of an interactive scheduler (consulting to other users).

- Distributed interpretation, which consists of the analysis and interpretation of distributed data; some systems of this kind are distributed sensors networks (Lesser & Erman 80).

- Distributed control and planning, which implies the coordination of the actions of distributed nodes to achieve the required actions. The main application example is distributed air traffic control (Finder & Lo 86).

The most interesting area in which we could frame our system is that of cooperating expert systems. These systems are based on the development of interacting collaborative mechanisms that allow the resolution of a common problem by means of many expert agents working

* This work is partially supported by EEC project COMETT II 90/1/1485/Cb

together. Because of the heterogeneous characteristics of each of these cooperative agents (due to the high specialization needed in a concrete knowledge area) each of them will be able to have different evaluation criteria to find the solutions, different ways of criticizing the problem and even different goals. In our system we try to integrate the different criticisms given by each of them to find a compromise solution in a cooperative way.

A common aspect to all of these collaborative design systems is that the designed architecture and the organization of the knowledge sources try, in some way, to simulate the real design process, where most of the time a cooperation takes place. This aspect is one of the motivations for the appearance of this new area in AI.

Taking into account these ideas engineering design problems are very adequate to be solved by collaborative systems. In the following we will show four different systems specially interesting and built upon different architectures (MacCallum & Carter 91), (Brown & Chandrasekaran 88), (Hunt & lee 92), (Myers & Pohl 92).

The first of them states that the coordination of the design relies on three main factors. The first one is the organization and control of resources or knowledge sources which are made by means of a blackboard architecture where the knowledge source are organized in a hierarchical form, following the real design process. Each of these knowledge sources has a specific knowledge. In the second place, to achieve resources integration, communication between knowledge sources through a common global area, is established. Finally there is a description and design distribution language, which is common to all the knowledge sources.

The second design system (Brown & Chandrasekaran 88), is a collaborative system because it tries to split a knowledge body in a hierarchy of specialists which cooperate between them. This organization is followed of the way to solve a problem by an human solver. The organization and control of the system is supported by a distributed top-down architecture to solve routine design problems. This design is made by a hierarchy of specialists which refine the problem when it passes down through the hierarchy. Each of the specialist uses a collection of design plans to construct the part of the design which is responsible. The communication is by means of message passing up and down in the hierarchy.

The previous collaborative design systems are systems which create designs. Our system, as mentioned above, uses the collaborative schema in the evaluation of the design. There are precedents in this kind of evaluation systems, for example a knowledge-based design assistant (Hunt & Lee 92). These refuse the idea of a completely automating of the design process; instead of it, they try to automate some design subtasks, which make available to the expert through existing CAD environment. The automatic evaluation is structured in three steps. In the first one they identify a similar design to the proposed design by means of a case-based reasoner. In the second one a functional model is used in order to obtain the suggested modifications for those functions of the design that do not match the requirements and finally a structural model is analyzed with respect to the original requirements. The authors suggest that this task is extremely complex and an human expert should perform it. It is just the main task of our system and without the need of an human expert.

Some recent works are trying to define the research topics best suited for Artificial Intelligence in design. (Schön 92) finds a promising goal in the construction of systems acting as assistants of human designers. Emphasis is placed in the enhancement of the designer's abilities. The system should also help in the recognition and reflection upon the consequences of his/her design actions.

In this line we find works like ICADS (Myers and Pohl 92), a cooperative multiexpert advisor system that monitors the evolving design process and helps in the resolution of some conflicts found in architectural design applications. Several problems appear when the solution becomes more subjective or when the capabilities of the system are extended, since the conflict resolver must know in advance all the conflictive situations and how to derive compromise solutions. The answer to some of the problems seems to be in more general and flexible coordination strategies.

In order to compare our system with the four systems showed before, we could say that the control of the knowledge sources is also supported by a blackboard architecture, but the resources organization and coordination is distinct because the application domain is different and also the form of the design, as we can know in each moment of the time that knowledge sources invoke; also, we realize of the cooperative nature of the system due to its structure and specially because of the established communication between the agents or knowledge sources, which can be compare with a debate or a negotiation between the experts, until getting an integration criteria.

In this paper, we expose a general architecture for a certain kind of design activities, namely the evaluation and refinement of a given design. In sections 2 and 3 the architecture is defined; in section 4 control issues are discussed; in sections 5 and 6, the implementation of this architecture in the domain of tutoring green areas design is described and finally some conclusions and future work are mentioned.

2. SYSTEM ARCHITECTURE

The goals of the system are two: the evaluation of designs given by the user and -when necessary- the suggestion of suitable modifications. Starting from a symbolic description of the design, these tasks are collaboratively carried out by a set of agents or modules. The general architecture of the system is displayed in figure 1.

The *blackboard* is the central element in the system. Several agents access the blackboard:

- A design problem is selected and displayed to the user by a *problem generator*. The problem is picked out from a prototype database comprising several designs. Obviously, the full design is not displayed to the user. Only certain graphical information and a verbal statement reflecting the main constraints of the problem are displayed (*problem statement*). The main goal of this prototype retrieving is to circumvent the very difficult problems arising in intent and plan recognition.

- The initial design is generated by the user -student or novice designer- in a *CAD environment*. A translation tool automatically yields the corresponding symbolic description in a design description language.

- *Design agents* analyze the initial design. Each agent is a specialist in a field and gives its opinion about the user's solution. The more general structure of an agent is displayed in figure 2.

- A privileged agent is in charge of the *coordination and blackboard control*, in the way explained below.

These agents read and write messages in the blackboard. The blackboard is made out by three main components:

1) *Symbolic description of the design.* An object-oriented symbolic language must provide a tool to model the objects and entities created through the design process. The design can be seen as a semantic network whose nodes and arcs are design elements and design relations, respectively. Every design agent has a particular view of the description, formulated in a *design description language dialect.*

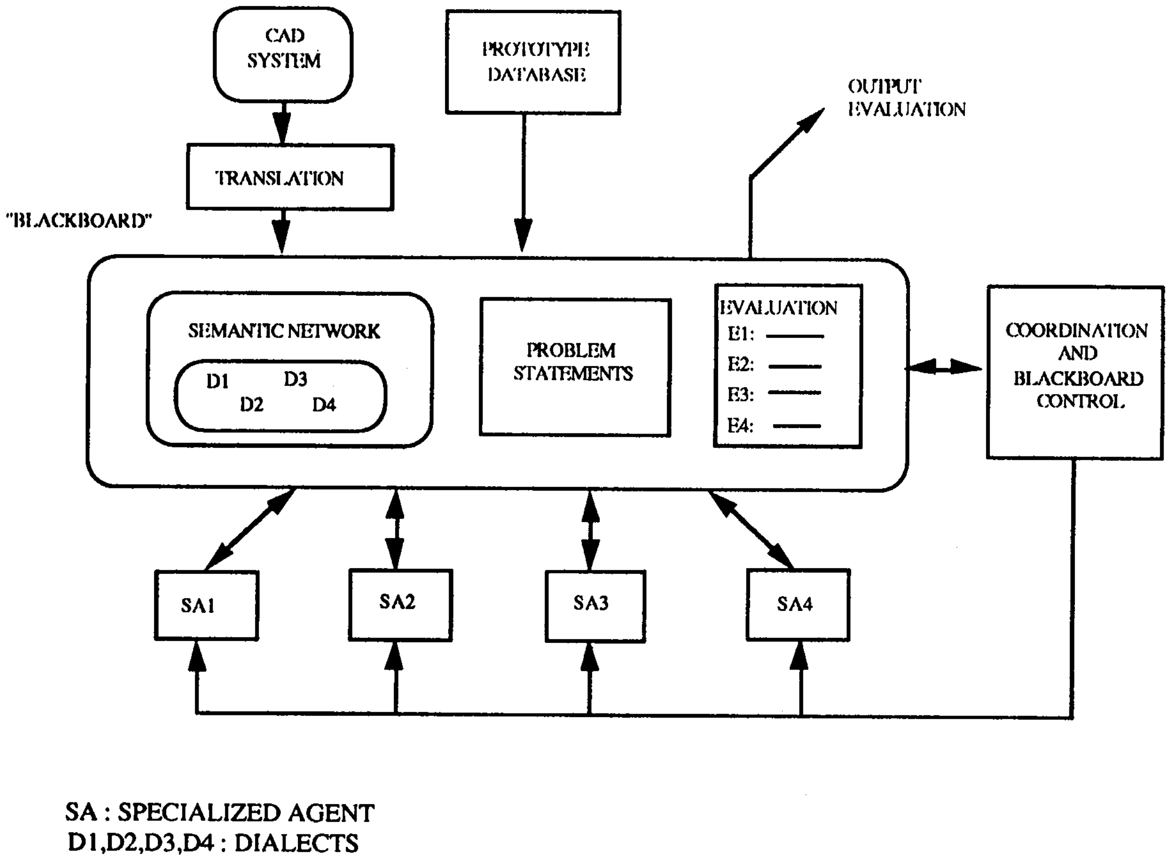


Figure1. System Architecture

2) *Symbolic description of the problem constraints and prototype.*

3) *Evaluation sheet.* Reports of design agents are stored in this area. Generally speaking, any agent can read reports generated by any other agent.

3. LANGUAGES FOR DESIGN REPRESENTATION AND COMMUNICATION

In order to carry out the symbolic processing of a design we need an adequate representation of the design. In our case, we analyze a particular class of designs, namely those involving spatial configuration of elements (objects). A good solution seems to be the use of a formal description language.

The proposed model relies heavily on the specification of a domain-dependent language, which describes the objects that form a design and their relationships. In other words, the language should be expressive enough to provide a complete description of a given design.

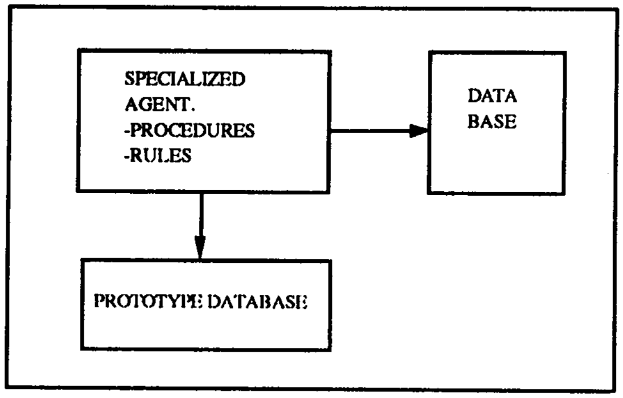


Figure 2. Functional Scheme of an specialized agent.

We will now consider how the specification of this language relates to the main ideas in our multi-agent architecture and turns out to be a key component of the system.

The evaluation of a design requires the collaboration of several specialized agents, which contribute with their particular domain knowledge. Each one has a set of rules and/or procedures suitable for the interpretation of different properties and relationships in the design. A single agent is therefore interested only in a subset of the whole design description (what we call a "dialect").

In some cases different agents will be able to read the same property or relationship, but always from different "points of view" (assuming the knowledge is not duplicated). When this happens, an agent might invalidate or agree with the other's statements, or otherwise a conflict requiring a compromise will be reached.

The existence of this "meeting points" (the intersection between dialects) defines a fundamental part of the system, the *predicates* that allow inter-agent communication. Different agents can argue about common predicates. These will be part of the description language and, therefore, particular of the kind of design we are considering. In some cases we will be talking about values of design variables, in others about object's relationships, etc.

We believe that the use of a domain-dependent language accomplishes several important functions:

- provides a symbolic representation of designs,

- forms the basis for a communication language between agents,

- the propositions of the language are the kind of facts needed by expert systems, or include values for procedure invocations.

We now directly address the problem of inter-agent communication, following these basic principles:

- each agent should be able to produce a complete evaluation (from its point of view) as if it were an independent system.

- communication should take place mainly in terms of the description language.

Our goal is not the search for optimal solutions or tradeoffs for every conflict. This task should be responsibility of the human designer. Our intention is to help the designer in the process of discovery and reflection upon the conflicts, point out inconsistencies and issue suggestions.

The output evaluation of each agent should be given in expressions like the following:

inadequate (Proposition, Reason)

change (Proposition, Suggested_Proposition, Reason)

comment (Object, Restriction_found)

...

In the first case, 'Proposition' is an object's property or a relationship between objects that has been rejected, and 'Reason' is a natural language explanation of why it was rejected.

This expression can be issued along with another like the one shown in the second place. This time the change of some part of the design is suggested. 'Proposition' is the fragment in the description the agent would like to change, 'Suggested_Proposition' is an alternate language proposition, and 'Reason' is a natural language explanation of why the change should take place. If several agents issue changes over the same proposition with contradictory effects, we are facing a conflict.

Other type of evaluation is aimed at focusing the designer's attention on some element (Object) which has certain limitations (Restriction_found) that should be taken into account before going ahead.

Finally, the coordination module should be able to handle individual and global evaluations, and issue an output report along with its justification.

An interesting outcome of this approach is that both, the human designer (thanks to the translation mechanism included in the graphical interface) and the specialized agents, talk about the design in terms of the same language. Thus, each agent can understand and criticize the propositions of the human designer and those suggested by other agents, as will be shown in the next section.

4. COORDINATION AND CONTROL

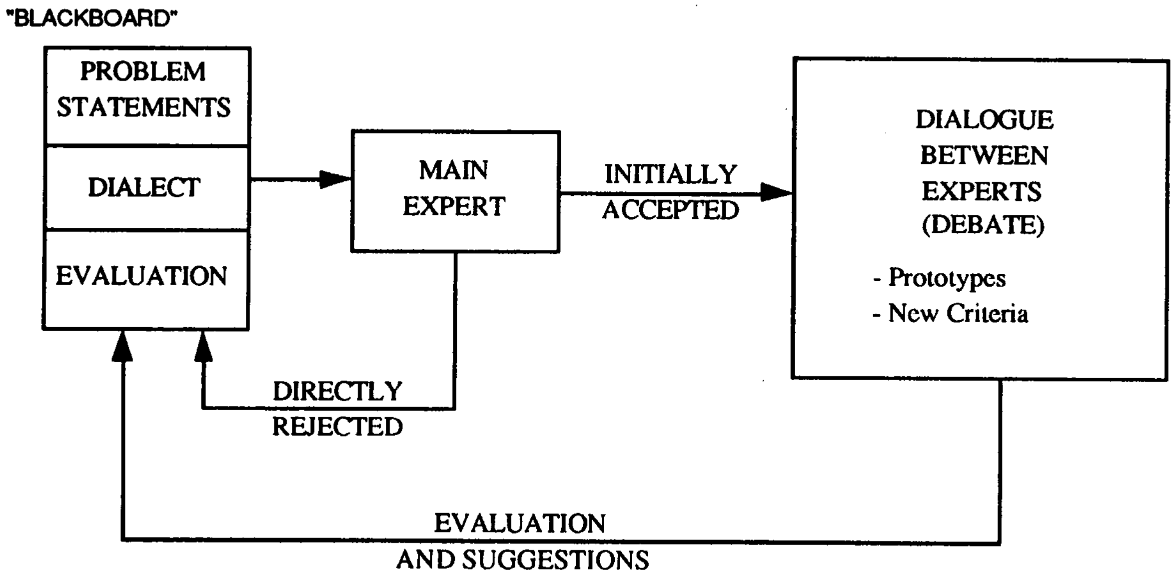
This module of the architecture, called "coordination and blackboard control", will be the responsible of the coordination and control of the distinct agents which evaluate the design.

Traditionally in the blackboard architecture designed for control, the main problem was to know in each moment of the time agent which should act and what actions should be made by it (Hayes-Roth 85) (Hayes-Roth 93).

In our system this problem is simplified because the intelligent part of the system does not make a design, instead of that it makes a evaluation without the need of actions, and in each moment of the time it knows exactly the specialized agent that will be invoked, called main agent, because it is clear depending on the actual stage in the design of the problem, as we will be shown later Sometimes it will be necessary to invoke several agents, establishing a communication between them to integrate their reports.

When the blackboard gives the control to the main agent to evaluate the design, this provides some criticisms; afterwards a round of debate is started between the remaining agents or experts. This round will be lead by the module of "coordination and blackboard control".

When different agents are reporting on a given design, it is very likely that their reports and evaluations are inconsistent or at least chaotic. So, a coordination module (*coordinator*) is needed to obtain an understandable consultation. A useful metaphor to structure the coordination is that of the debate (see figure 3).



DIALECT : Portion of the full design description related to a specific issue.

MAIN EXPERT: Specialized Agent most related to that specific issue.

Figure 3. Evaluation Process

The coordinator leads the debate. The evaluation is divided into a sequence of issues. For each issue, the coordinator selects an agent as the main expert. The main expert reads and evaluates its dialectal design. If the evaluation output is not definitely unacceptable, a round of debate is started, in which the remaining agents put remarks on the evaluation sheet. Then a new issue is proposed by the coordinator, so beginning a new round of the debate. When the consultation is over, the evaluation sheet stores a network of remarks where conflicting opinions may be reflected.

Starting from the messages stored in the evaluation sheet, it is possible to generate a consistent report where pros and cons of the given design are considered, and better alternatives are suggested.

5. APPLICATION DOMAIN

The above described architecture has been applied in a system for tutoring the design of green areas. Actually, green areas are very complex objects, and green area design is closer to the "reflexive conversation" of (Schön 92) than mundane mechanical design so often found in this context. A first simplification of the design situation is carried out by considering only design situations originated by *problem statements*.

However, there are many aspects to be taken into account, ranging from the aesthetic to the botanical ones. The natural way to implement this variety of involved disciplines is by using a multiagent collaborative architecture like the one described in this paper.

There are usually four stages in the design of a green area (González and Cañizo 88):

-1st draft. The area is divided into several zones, and the functions of each zone are defined.

-2nd draft. Architectonic elements are placed into the area.

-3rd draft. Botanical elements are specified according to form and function.

-4th draft. Suitable botanical species and varieties satisfying the constraints specified in the3rd draft are selected.

Each stage has a main criterion associated with the decisions taken:

-Functionality
-Functionality/aesthetics
-Aesthetic/botanical
-Botanical.

Each of these criteria will provide one specialized agent or expert (implementation modules of each criteria):

- Functional Agent: This agent will evaluate the distribution of the parts in the green area (for example situation of the house, sitting area, play children area, garage...) and the correct distribution of the different species of trees and bushes with a functional point of view (for example the correct position of the barrier of trees situated in the green area to get isolation from the exterior). It also will adjust sun areas and shadow areas (for example the sun is most important in an Atlantic climate than in Mediterranean one because the Atlantic climate is colder than Mediterranean). Finally it will evaluate some aspects of functionality of the path, as shape or width.

- Botanical Agent: This agent will evaluate that the species of the plants are correctly planted and it verify climate, ground and altitude restrictions; and also the correct measures of the plantation of trees and bushes. For these measures are important some aspects as dangerous roots, and the width of the top.

- Aesthetic Agent: This agent must keep up the style of the green area, trying that all the components of the green area have coherence or a similar style (for example Natural garden, geometric garden). It also will evaluate the coupling between the components of the green area because a green area must be designed as an entity. Finally it will check that the main visual axes of the green area, as views from the windows or privileged points, coincide with attractive architectonics or vegetal elements.

For obtaining more objectivity it will be interesting that this agent gives its opinion about established valuation levels, as chromatic variety, shapes variety, florist richness, aroma...

In this scenario, the proposed architecture allows the collaboration of modules implementing each criterion, namely a botanical agent, an aesthetic agent and a functional agent.

On the other hand, the user -the student- designs in a structured CAD environment where his actions are automatically displayed and translated into a symbolic *'green area description language'*. The design is motivated by a problem statement where relevant terrain and environment features and weather conditions are set out.

When this proposal is finished, the user invokes the main module of the system. The coordinator takes the control and in turn each of the above mentioned agents assumes the role of the main agent and analyze the symbolic description of the design. Finally a short tutorial suggestion is displayed to the user, for example, "The plants selected are pretty good, but perhaps it would be better to isolate the west border with taller trees".

6. A LANGUAGE FOR THE REPRESENTATION OF GREEN AREAS

A language for the description of green areas has been implemented. This language is used to generate the representation of the designs drawn by the user in the CAD environment and for communication between the specialized agents (functional, aesthetic and botanical).

This language includes the objects and relationships that define a green area and that are necessary to produce an evaluation. There are three main classes of objects in the designs: zones, architectural elements and botanical elements.

The partition of the land in "zones" is made according to functional criteria. Each zone is characterized with several properties, such as location, functionality, climate and kind of soil. They also contain the necessary architectural and botanical elements, characterized with their own properties.

An essential property is the species of the botanical elements, which allows the retrieving of additional information from plant databases. This kind of information is not explicitly stated in the design process, but forms part of the designer's knowledge and is of great importance in the evaluation process.

Finally, the representation is completed with a topographical description of the land.

Here is an example of the use of the language. We will consider a potentially conflictive situation. Let's assume that the south border of a small piece of land with Atlantic climate is adjacent to a dumpsite (represented internally as a "bad view"). The aesthetic agent will immediately suggest a tall compact hedge that isolates our garden from such a terrible view. However this suggestion will be read by the functional agent, who will in turn point out that the hedge would block the sunlight, leaving the garden in shadow (it is placed in the south border). The designer will have to choose between a bad sight and limited sunlight.

7. CONCLUSIONS

In real world, design processes are usually collaborative, and that is a good reason to use a multiagent architecture for automatic design systems. We strongly believe that a true tutorial system must also reflect this feature, especially when interdisciplinary tasks -like green area design- are taught.

A general domain-independent architecture supporting collaborative evaluation and refinement of design has been defined. This architecture is currently being implemented in the domain of green area design. We have obtained some encouraging results that point out the feasibility of the task and the adequacy of the architecture. We are now addressing more substantial problems in this domain, and in a near future we hope to develop similar tools for other engineering areas.

REFERENCES

Bond and Gasser (1988). *Distributed Artificial Intelligence.* Morgan Kaufmann.

Brown and Chandrasekaran (1988). *Expert Systems in Engineering.* IFS Publications/Springer -Verlag, Pham, D.T. (eds.).

Chaibdraa,B. and Millot, P. (1990). A framework for cooperative Work: an approach based on the intentionality, *Artificial Intelligence in Engineering.*Computacional Mechanics Publications.

Croft, W.B. and Lefkowitz, L.S. (1988). Knowledge-Based Support of Cooperative Activities, *Readings in Distributed Artificial Intelligence* : 599-605.

CSCW (1990). *Proceedings of the First Conference on Computer Supported Cooperative Work.*

Durfee, E.M., Lesser, V.R. and Corkill, D.D. (1989). Cooperative Distributed Problem Solving, *The Handbook or Artificial Intelligence IV.* Barr, A., Cohen, P.R. and Feigenbaum, E.A.. Addison-Wesley. Reading, Ma.

Engelmore, R. and Morgan, T. (1988). *Blackboard Systems.* Addison -Wesley.Reading, Ma.

Gonzalez,R. and Cañizo, J.A (1988). *Jardines: Diseño, proyecto, plantación.* Mundi-prensa Ediciones. Madrid.

Hayes-Roth, B. (1985). A Blackboard Architecture for Control, Readings in Distributed Atrificial Intelligence : 505-540.

Hayes-Roth, B. (1993). Intelligent Control, *Artificial Intelligence* **59**: 213-220.

Hunt, J.E. and Lee, M.H. (1992). Towards a Knowledge-based Design Assistant, *Engineering Application Artificial Intelligence* **5**(4): 275-287.

Leong, K.T. and Sim, S.K. and Chan, Y.W. (1991) BESMED: A blackboard Knowledge-based Aproach to Integrate Mechanical Design, *Engineering Application Artificial Intelligence.* **4**(3): 205-220.

Lesser and Herman (1977). A retrospective view of the Hearsay-II architecture, *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)* : 790-800.

MacCallum, K.J. and Carter, I.M. (1991). Opportunistic software architecture for support of design coordination, *Artificial Intelligence in Design'91* Conf.Edimburgh,UK.

Myers, L. and Pohl, J. (1992). ICADS expert design advisor: an aid to reflective thinking, *Knowledge-Based Systems* **5**(1): 41-54.

Schön, D.A.(1992). Designing as reflective conversation with the materials of a design situation, *Knowledge-Based Systems* **5**(1): 3-14.