2.1
Conversation on Design Action: By Men or by Machines?

Ivan Petrovic and Igor Svetel

Bussiness adress: IMS Institute, POB 834
Belgrade, Yugoslavia

A design studio of the future shall be based on dislocated, distributed design services, and feature the 'design by collaboration' enabled by the computer transmitted information. However, in a collaborative design process, computer may take an additional role, i.e., as an "ultimately structured dynamic communication medium ... based on the notion of commitment and interpretation" (Winograd and Flores 1987). Various models of 'intelligent' design systems based on the ideas of 'open, distributed, artificial intelligence systems' have shown that the computer-based design agents which act on the object-to-be-designed model could be involved in a "conversation for action" (Winograd and Flores, Ibid.). The aim of the paper is to illustrate a computer-based design system that enables 'a-kind-of' conversations by the design agents before the design decisions were made. After the description of a design experiment and the conversation that went on between the design agents, the traits of the applied 'design design system' are discussed.

Introduction

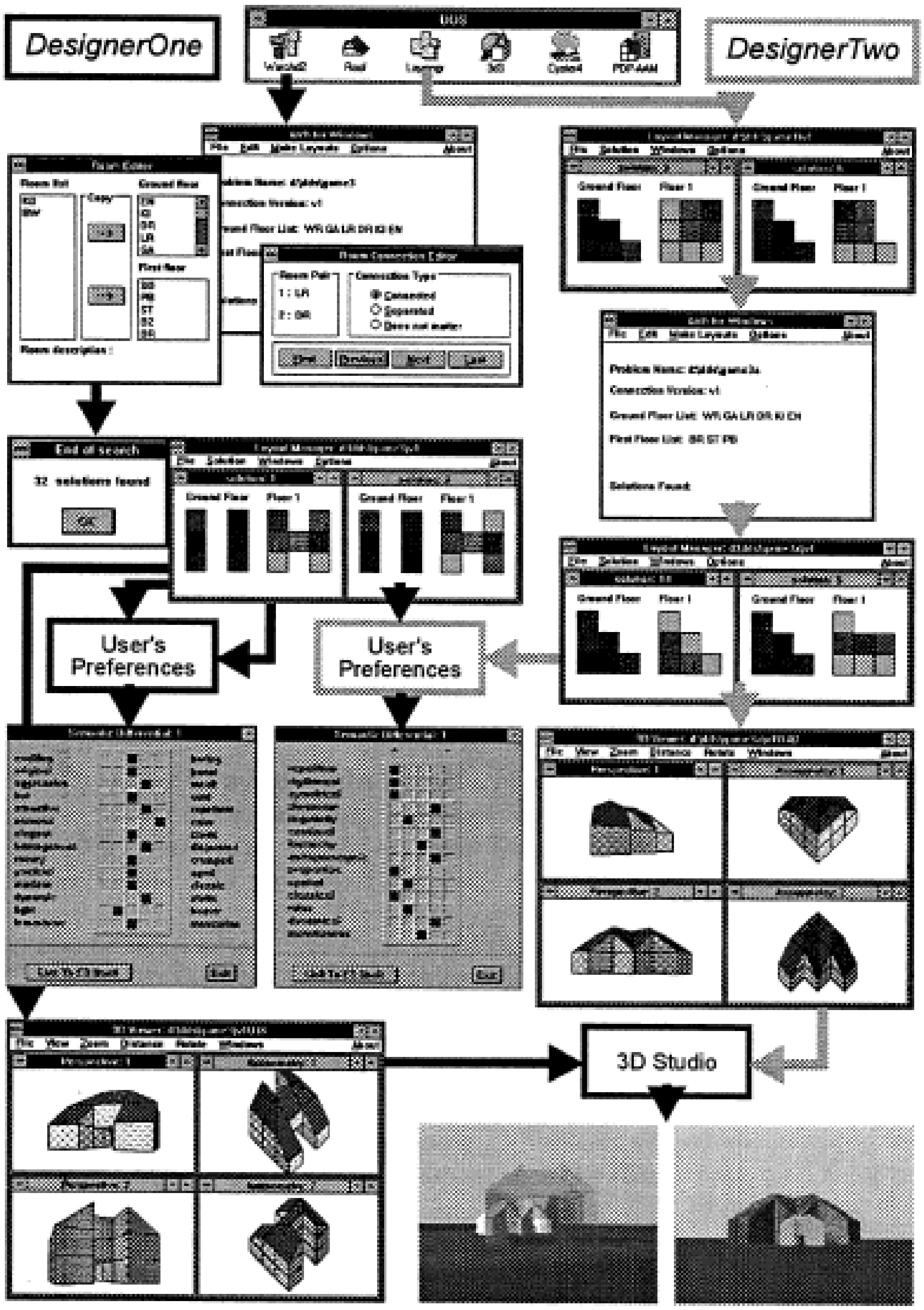
A design studio of the future shall be based on dislocated, distributed design services, and feature the "design by collaboration", enabled by the computer transmitted information (Mitchell 1994). However, in a collaborative design process, a computer may take an additional role, such as an "ultimately structured dynamic communication medium ... based on the notion of commitment and interpretation" (Winograd and Flores 1987). Various models of 'intelligent' design systems (Pohl et al. 1992; Fisher and Nakakoji 1992; Rutherford and Maver 1994; Petrovic 1994; Svetel 1994), based on the ideas of 'open, distributed, artificial intelligence systems' (Gasser 1991; Hewitt 1991), have shown that the computer-based design agents that act on the 'object-to-be-designed' model could be involved in a "conversation for action" (Winograd and Flores, Ibid.).

The aim of the paper to is illustrate a computer-based design system that enables 'a-kind-of' conversation by the design agents before the design decisions were made. The proposed 'Design design system' is, as a whole, a collaborative, anticipatory, non-deterministic enterprise. It is comprised of design agents, which behaviours (modalities of work and use of tools) reflect the traits of the design process.

An Experiment: Conversing About, During and After a Design Action

This paper is about a specific computer system (Figure 1.) and its behaviour (Table 1.). We think that the best, and probably the only way to explain the traits of a system of this kind is to try it hands-on, see what it does and how it works, and only then to learn about it. After an experiment, the questions' one asked oneself would be as follows: ‘What was the aim of the experiment?’, ‘How to check the results?’, ‘Are the results repeatable?’ etc. The answer to the last questions is that each experiment would, most likely, be a unique one. This is because the aim of a ‘conversational’ system is primarily be the exploration of a design situation. Its performance would very much resemble to the work of a traditional architect with ample free time, just trying to ‘feel’ the design situation.

Figure 1. An Exploratory Design Session: Design of a Gate House



The following description of a design session refers to a short experiment that was not prepared in advance. The initial DDS screen shows the existence of 6 ‘design agents.’ We also consider the humans the design agents. Some agents are ‘generators’ (i.e., generate house layouts and roofs), others are ‘evaluators’ (i.e., evaluate layouts or complete buildings from the specific aspects), while other agents perform some specific functions.. The names of the computer agents are: ARCH (A Generator), LayoutManager (A Data manager), 3d Viewer (A Transformer of data representations), RoofMaker (A Generator), OYSTER (An Evaluator), and PDP-AAM (A Generator - Evaluator) (Table 2.)

DesignActionOne. A human agent *DesignerOne* has decided to explore some alternative possibilities of a design of a ‘Gate’ GIMS house. Inasmuch the house obviously implicates a certain meaning to the *DesignerOne*, it is also a house with standard housing functions. Therefore, the *DesignerOne* informs the *ARCH* about the house content (types of rooms) and their relationships.
DesignActionTwo. *ARCH* starts with the work, and sends the message to all agents when it has completed the production of the alternative designs. *DesignerOne* may wait to see the first proposal that *ARCH* produced, or may prefer to do something else in the meantime: e.g., view some previous solutions, or activate any of the other agents.
DesignActionThree. After its activation, the *LayoutManager* presents solutions on the screen. *DesignerOne* can view all of the produced alternatives in the selected mode of presentation (e.g., one-by-one or in a multiple view), and make some preliminary off-hand selections. However, other agents are also offering their help. For example, *OYSTER* has an expert knowledge-base that can evaluate the rooms orientations and their relationships to the outside. However, other *OYSTER* knowledge-bases may contain the clauses from the housing regulations, or the knowledge-bases by other people containing the different views on the same topic - orientation. For example, the *DesignerThree* is known for his preference to open living-rooms to the north, while the *DesignerTwo* is known for her opposite preferences. The *DesignerOne* can in this way ‘converse’ with other agents about this topic, and accept any view at will.
DesignActionFour. In the meantime, the *DesignerTwo* has decided to start exploration with a ”big” house, and, concurrently with the action of *DesignerOne*, activates the *LayoutManager* for viewing of the existing "big" houses. One of the solutions ‘inspired’ the *DesignerTwo* to explore the possibilities of a house with a pyramidal form. The *DesignerTwo* sends the message to *Arch* ordering the types of rooms and their relationships of a new house. Soon, *Arch* produces the expected design alternatives, and the *DesignerTwo* calls *RoofMaker* to see what roof alternatives are possible. *RoofMaker* finds only one roof. The *DesignerTwo* tries to investigate formal properties of the building, and for that reason *PDP-AAM* is called in order to produce the semantic differential that accounts for the appropriate (‘formal’) design traits.
DesignActionFive. By now, *DesignerOne* is viewing the produced 3D models of the design alternatives, the roof alternatives, and the corresponding semantic differentials of the Gate house. During the viewing, *DesignerOne* becomes curious to know what the *DesignerTwo* might think about the offered solutions of the Gate house. The *PDP-AAM is called, only to show that* that the *DesignerTwo* has recently produced new semantic differential for house evaluation, based on the formal, visual properties of the house forms, such as symmetry, proportions, rhythm, etc. *DesignerOne* checks the solution of the Gate house against the *DesignerTwo's* preferences. This done, the *DesignerTwo* calls back the *LayoutManager* to see *DesignerTwo's* solutions, and proposes to the *DesignerTwo* that both designers make a try and propose a composition comprised of both buildings.
DesignActionSix. Both designers produce the DXF files that represent 3D models of the buildings, and leave GIMS-DDS. They loads 3D Studio (3d Studio is a Trademark of Autodesk Inc.) in which they make different compositions of the houses, experimenting with change of their dimensions, relationships, materials, etc.

Table 1. ‘Protocol’ of a Design Session Experiment

The focus of this experiments was on the design process and not on the results. This can be easily proven by examining the achieved 'design quality' of the houses which would, most likely, need further design refinement. At present, the proposed design system is applied to 'GIMS houses' (Petrovic 1982). We have deliberately limited the design domain to relate only to the sketch design of small houses contained within a two-storey structure based on the three-by-three square design module. We consider it sufficiently large for our initial investigation because we are interested not only in agents conversation, but also in other methodological aspects of the system.

ARCH. A Generator. Receiving information (input): Room-list, Connections list. Communication type (during action): Sending non-wakening-up (informing) message: completion of a new design. Sending information (output): Layouts list.
LayoutManager. A Data manager. Receiving information (input): Layouts list. Communication type (during action): Sending wakening-up (requesting) messages: Consultation (orientation evaluation, subjective evaluation, request for making 3D representation of objects. Screen: House layouts
PDP-AAM. The roles: Generator, Evaluator, and Transformer of semantic differential requests into a 3d representation, and Transformer of 3d representation into the respective semantic differential. Receiving information (input): (1) Semantic differential, (2) 3d Building representation. Sending wakening-up (requesting) message: Consultation. Information (output): (a) 3d Building description, (b) Semantic differential. Screen: (A) Semantic differential, (B) 3d Building description.
OYSTER (Client-Mode). An Evaluator. Receiving information (input): Facts-Base, Knowledge-Base. Communication type (during action): Wakening-up (requesting) message: new facts during the backward chaining.. Sending information (output): Facts-Base, Knowledge-Base. Screen: Facts-Base, Knowledge-Base.
OYSTER (Server-Mode). An Evaluator. Receiving information (input): Facts-Base, Knowledge-Base. Communication type (during action): Receiving wakening-up (requesting) message: Consultation. Sending information (output): Facts-Base. Screen: Facts-Base, Knowledge-Base.
RoofMaker. The role: Generator. Receiving information (input): 3D Building description (without roof). Communication type (during action): Sending information (output): Alternative 3D House description (with roofs). Screen: none
LittleModeller. The role: Transformer of data representations. Receiving information (input): Building layout. Communication type (during action): Receiving wakening-up (requesting) message. Sending information (output): 3D Model. Screen: none.
3D Viever: The role: Transformer of data representations. Receiving information (input): 3D Model. Communication type (during action): Receiving wakening-up (requesting) message. Sending information (output): Screen: Building perspective view of a building.

Table 2. List of DDS agents

Design Agents at Work: Conversing About Design Action

During the experiment, an observer would quickly recognize that the communication between design agents resemble to a 'conversation'. The Table 3. presents a preliminary framework for classification of the main types of the design conversation during a design action, based on types of speech-acts proposed by Winograd and Flores (Ibid. p. 59). The communication information between the design agents may be considered to be:

(1) Assertive, committing another agent to something's being the case - to the truth of the expressed propositions. Examples: the traits in the semantic differential.
(2) Declarative, (bringing about the correspondence between the propositional content of the design action and reality). Examples: building of the facts list about room orientation (e.g., LayoutManager preparing factlists for OYSTER).
(3) Directive, such as one-way commands, or 'messages'. Examples: *accept, reject, engage, select, withdraw, count, stop, show, number of design alternatives, warnings, wake-up calls.*
(4) :***Committing, e****ntrusting* a design agent to some future course of action and interested in the reply. Examples: *generate, evaluate.*
(5) Expressive, expressing a psychological state about state of affai such as 'liking' or 'disliking' a house etc. Examples: conclusions inferred by OYSTER.

Table 3. Types of Communication Between Design Agents

The following types of relations between the sender and receiver were employed: (a) one-way and (b) two-way, and (1) one-to-one, (2) one-to-many and (3) many-to-many.

All agents described belong to the known variety, except for the *PDP-AAM.* This programme is based on a learning, 'neural net' model, employing the back-propagation algorithm (Rummelhart et al. 1986). Its interface contains: (a) a semantic differential, allowing a description of a house, and (b) an editor for the 3d representation. This agent enables the following performance: (1) definition of a semantic differential for the given house, and (2) proposal of a house for the given semantic differential (Svetel 1993).

Let us examine for a moment the following examples. When ARCH sends a message 'I have made thousand design alternatives (or 'design solutions'), would you like to see them!" is certainly a conversational remark. When OYSTER represents some house semantic differential traits and finds that 'The house is right for the user in the given context!' is also an utterance which puts some label to the house, placing it into a certain category. Obviously, the human designers also converse exchanging knowledge-bases and semantic differentials, issue commands, send data and listen to the advises given by other agents. More such examples could be extracted from any newly performed experiment.

The Design System

The DDS is a distributed system resembling to Minsky's (1986) 'open society' of the autonomous, cooperative design agents. The majority of agents represent well understood design procedures, but the system includes at least one human participant. The process does not have any predefined sequence of steps (design agents applications) and allows the start and finish at any point of the process. All agents are engaged in 'doing their thing', that is, in transforming the object-to-be-designed the best way they can. The system combines different knowledge representations trying to overcome the limitations inherent in each one. Also, it combines activities of many processes of diverse complexity and level of 'intelligence'. Concurrency is an inherent quality of the system, since the multiple parallel steps can follow the previous ones. The implementation environment is MS Windows 3.1 (Windows is the trademark of Microsoft Corporation), while the implementation language is Turbo Pascal for Windows 1.0 (Turbo Pascal is the registered trademark of Borland International Inc.). The system is organized on three functional levels: (1) application and communication level, (2) knowledge level, and (3) data level.

A Sketch of a Design Design System Paradigm

'DESIGN DESIGN SYSTEM'	
Based on the phenomenological concept of design.	*Metaphors:* phenomenological, ideation, exploration, wholism, emergence, creativity.
Influence: hermeneutical philosophy, biology, neurology, social sciences, self-organizing systems.	*Instrumental basis:* colaboration, conversation, commitment, categorization, open world assumption.
Basic value systems: value judgments.	
System structure: circular, 'whirling' (Hickling 1982), asynchronous.	*System elements:* design 'agents'.
Some agents performances are only anticipatory.	Man is a part of the design system.
As the designers - users are parts of the system, the 'participation' is reality, "allowing experts to relate their actions, even by constantly changing pattern of the problem" (Jones 1970 p. 20).	After the system is completed, it can be reassembled and modified in such a way that a variety of new uses may appear which were not envisaged at the system conception.
The system includes 'a variety of 'design tools'.	Control of design decisions is *inside* the design system.
Design may start exploring particular design requirements, but in principle, it does not 'start' but 'continues'.	Existence of design objective is possible but not essential.
Design is treated as a whole, but the individual design states may be used independently.	The system is 'open' for information exchange during the process.
The system employs both, the problem-solving procedures and the procedures that may not lead to conclusions.	Design methods are 'societies of computer-aided design agents'.
Design application domain: human design aspects.	
Present application in architectural practice: ?	*Application in architectural education:* as research subjects, and demo programmes.

Table 4. Some Traits of a Design Design System

Application and communication level contains all DDS agents and communication processes. The basic communication between the agents is represented as a set of the specific messages that correspond to the particular actions of the agents. Large amounts of information are transferred using a set of messages coupled with the query keyword, informing an agent where it can find the appropriate information.

In addition to these communication protocols (1. message, 2. files + query key words) the system implements a central blackboard to hold global data that affect all agents in the system. The DDS blackboard contains data structures holding relevant global data, and a collection of methods for storing, retrieving and deleting data that can be used by all DDS agents. Data stored in the blackboard describe the current DDS configuration.

The knowledge level accommodates all knowledge sources that can be applied during the design process. Its structure is distributed and heterogeneous. The current DDS implementation contains a rule-production system interpreted by OYSTER shell, geometric pattern-directed production systems interpreted by ARCH, and RoofMaker, LayoutManager, and a connectionist network weights interpreted by PDP-AAM. All knowledge sources are realized as the separate files that establish the particular pieces of design competence. Moreover, during the design session, a completely new knowledge base may be added, under a different name. It is best if it is of a different nature from the existing ones, because it may interfere with already formed conclusions. The changes of the existing knowledge-bases ones are possible but their maintenance has not been provided at this time.

The data level contains all information describing the design at hand and its alternatives. The data structure is also distributed as a collection of files that represent parts of the topological, geometric, descriptive and factual data. They are organized in a top-down hierarchical directory structure. This structure serves as a query keyword that permits an agent to find the appropriate data in a distributed data-base. The keywords are communicated between agents informing them about the appropriate data necessary to establish their activity.

The accepted organization of the DDS system is a ‘pluralistic community model’. The model was devised as an analogy of the behaviour of a scientific community (Kornfeld and Hewitt 1981). Each DDS agent is capable of treating a design issue from a particular point of view. Since the agent's knowledge displays the 'chunks' of specialized design judgment, it is possible that agents have the disparate viewpoints of the problem solution. For this reason, all results are considered as beliefs and are advanced to other agents for further contemplation. The global ‘solution’ is achieved by enlargement of the supporting evidence for design alternatives through successive iterations.

Conclusions

The formation of the ‘DDS’ was the result of a follow-up of the previous, practice-oriented research in computer-aided design (Petrovic 1994), and of the exploratory development of a design system that shares ideas with the studies of relationship between phenomenology and design (Coyne and Snodgrass 1991), distributed artificial intelligence (Gasser 1991; Hewitt 1991), and the similar research projects in the field (Pohl et al. 1992; Rutherford and Maver 1994; Fisher and Nakakoji 1992; Flemming 1993). In this sense, the DDS is an attempt to devise ways and means to use computers perhaps to the liking of ‘not-so-rational designers.’ The play-like features of the DDS makes it applicable in conceptual design stage explorations, or to a researcher in the further investigation in this field.

It is feasible that the DDS may be of the best use to such designers who incline towards explorations during the design process. It will be of less use to those who do not like to change their first ideas and are consequent in their decisions to the very end. Although the DDS can handle the well-defined situations, it is really not needed there. Nevertheless. some of the features of the DDS, such as the very quick development of alternatives and their visualization, may help a practical designer with little time to spare.

The DDS agents are models of the separate design procedures that can be implemented as independent processes. Consequently, each agent represents a distinct design process that can be engaged separately from other processes. The agent receives its input from other agents, produces the suggestion, and sends it to other agents which can improve, modify, appraise or criticize the proposed design. The functionality of the agent does not depend on the activity of other agents. In case of the failure of a particular agent, the activity of other agents would not be affected, only the scope of the services offered by the system will be reduced (e.g.,, some sort of design evaluation, or some representation of the evolving design will not be available at that moment). Cooperation in DDS is considered as the way to extend the scope of design judgment, taking into account various aspects of design thinking. We see the DDS as an emerging system which will evolve in two directions: (1) adding new agents, and (2) replacing the existing ones. We believe that the DDS environment will give birth to such new computerized design agents that cannot be forecast at the moment.

The exact definition of a ‘design agent’ is not so simple. We use design tools in such situations when the purpose of the action is known, and a search for the means to achieve the result is required. An agent is shearing the responsibility with us and is commited to do something that we can not. For example, it would be very difficult to interpret the semantic differential in view of many parameters, or generate hundreds of alternative designs on a single theme. The traditional designers who select and employ various methods and tools in their work are true design agents. However, the claim that the human designers are always design agents becomes blurred in such cases when a ‘chief designer’ tells another designer to do something on which the latter has no influence. Sometimes the computer design agent may issue orders to a human designer to do something, as it happens in case of expert systems.

Our attempt to isolate the main characteristics of a design conversation is obviously just the beginning of a more serious effort that should we continue the work in this field, and look for some new design agents. As for now, in a phenomenological sense, we ‘know’ that our agents exist, converse in one way or another, and produce some results. Working on this theme opened up a better way of understanding the way the design agents work. Nevertheless, we would like to point out that we consider the conversation between design agents as a means and not an end in design systems development!

We do not doubt in the further advance of the rational, computer-aided, problem-solving design systems, methods and tools. This approach shall account for a great portion of designer’s practice in the future. But there is no need to deny that the designers do not always select ‘methods’ in a rational way, just as their final designs are not always considered to be the most rational answers to the problem. Therefore, in some exploratory phases of the conceptual design, or in preparation for design, shall, in our view, always be room for a DDS-like adventure. A problem-solving design system cannot start without a ‘stimulus’, be it a ‘need’, ‘requirement’, ‘design brief’ or similar. But, why not experience the sometimes forgotten great mystery and joy of design and simply ‘start’ and see what happens in the end. Seredipidity has always been one of the design features highly regarded by certain designers: Even in the presented experiment, the result of the design session was quite different from one that had been expected. We believe that this was due to the conversation that took place between *all* agents involved in the design session.

What are the possible uses of the DDS in a virtual design studio? We believe that the DDS-like design lies in a parallel, virtual plane of any ‘real’ design. In future, we see more diverse design agents joining the present society, but with the increased ability to converse, thus creating more worry and responsibility to the human design agents. Similar design agents societies shall be indispensable in design education, another virtual plane of the design reality.

References

Coyne R. and A. Snodgrass. 1991. ”Is designing mysterious? challenging the dual knowledge thesis.” *Design Studies* 12, no. 3:124-131.

Fischer, G. and K. Nakakoji. 1992. ”Beyond the macho approach of artificial intelligence: empower human designers do not replace them.” *Knowledge Based Systems* 5, no 1:15-30.

Flemming, U. 1993. ”SEED: A software environment to support the early phases in building design.” *Proceedings ARECDAO '93*. Barcelona. 111-122.

Gasser, L. 1991. ”Social conceptions of knowledge and action: DAI foundations and open systems semantics.” *Artificial Intelligence* 47: 107-138.

Hewitt, C. 1991. "Open Information Systems Semantics for Distributed Artificial Intelligence." *Artificial Intelligence* 47: 79-106.

Hickling, A. 1982. "Beyond a linear iterative structure" in Evans, B., J. Powell and R. Talbot (eds.), *Changing Design*. New York:John Wiley and Sons .275-294.

Jones, J.C. 1970.*Design Methods*. New York: John Wiley and Sons.

Kornfeld, W.A. and C.E. Hewitt. 1981. "The scientific community metaphor." *IEEE Trans. Syst. Man Cybern* 11 no. 1:24-33.

Minsky, M.. 1986. *The Society of Mind*. New York: Simon & Shuster Inc.

Mitchell, W.J. 1994. "Three Paradigms for Computer-Aided Design" in Carrara, G. and Y.E. Kalay (eds.), *Knowledge-Based Computer-Aided Architectural Design*. Amsterdam: Elsevier.

Petrovic, I. 1982. *GIMS houses and GIMS Method*. Belgrade:The IMS Institute Report.

Petrovic, I. 1994. "On Some Issues in Development of Computer-Aided Architectural Design Systems" in Carrara, G. and Y.E. Kalay (eds.), *Knowledge-Based Computer-Aided Architectural Design*. Amsterdam: Elsevier.

Pohl, J., L. Myers, J. Cotton, A. Chapman, J. Snyder, H. Chauvet, K.J. Pohl, and J. La Porta. 1992. *Implemented and planned extensions of the ICADS model*; Tecnical Report CADRU-06-92, San Luis Obispo: CAD Research Unit, Cal Poly.

Ruherford, J.H. and T.W. Mawer. 1994. "Knowledge-Based Design Support", in Carrara, G. and Y.E. Kalay (eds.), Knowledge-Based Computer-Aided Architectural Design. Amsterdam: Elsevier.

Rumelhart, D.E., G.E. Hinton and R.J. Williams 1986. "Learning internal representation by error propagation." In Rumelhart D.E. and J.L. McClelland (eds.), *Parallel Distributed Processing*. Cambridge: MIT Press 318-362.

Svetel, I. 1993. "PDP-AAM: An analogical architectural modeller." *Proceedings ARECDAO '93*. Barcelona.

Svetel, I. 1994. "DDS - Distributed Design System: A Paradigm for Modeling Conceptual Architectural Design." *Proceedings InterSymp'94*. Baden-Baden.

Winograd T. and F. Flores 1987. *Understanding Computers and Cognition*. Reading: Addison-Wesley.