A Framework for Cooperative Activities of Computer Design Agents

by

Ivan K. Petrovic

Institute IMS
Belgrade, Yugoslavia

Ivan K. Petrovic

is Research Director at the IMS Building Research Institute in Belgrade. Professor at the University of Arts in Belgrade. Vice-President ECAADE. Member CIB-W78 Commission. Designed dwellings in Belgrade, Sarajevo, and Luanda. Research in the field of design methodology and CAAD.

Abstract

The paper presents a progress report on a project investigating the possible application of a framework for cooperative-activities of computer design agents in the conceptual phase of architectural design. A process leading to definition of the expected performances of design agents is desribed, and some possible applications illustrated. The framework includes not only the objective, but also, the “subjective” agents. It is expected that the framework would offer an insight into the intricacies of CAAD in an educational environment, and provide the exploration paths and an efficient production of alternative solutions in an office.

Introduction

Can computers help in the conceptual stage of design process? How? In recent years many authors have investigated these issues and provided a multiplicity of answers. Our approach assumes the affirmative answer to the first question and proceeds with a proposal that the application of computers is organized through a loosely organized framework of “computer design agents” which, through series of cooperative activities, transform the model of the object-to-be-designed. Many researchers have found the distributed cooperative models of architectural design process advantageous in relation to the centralized systems [Pohl, 1994], featuring openness, modularity and quick adaptability to context changes [Rutherford et al., 1994]. What we expect to add to the present efforts is the reconciliation of the typical objective approach to design with the subjectively-based techniques that have become possible to implement by the new computational techniques [Coyne, 1990]. We find the support for our approach in the challenge of the view of design as “the product of the objective individual” [Coyne, 1991]. The development of the proposed design framework is based on our previous work of the similar nature [Petrovic, 1993; Petrovic, Svetel 1994a; Petrovic, Svetel 1994b; Svetel, 1994].

SELECTION OF THE MODEL OF DESIGN PROCESS

Introduction

One of the first tasks in our project was to decide what model of the conceptual phase of design process to use. Our idea of what constitutes the conceptual design is to a great extent based, in the phenomenological sense, on observation how designers actually work [Schˆn, 1992]. Regardless of the nature of design process, be it “rational” (comprised, for example, of a definition of desired objectives, production of alternative design solutions, and evaluation of the solutions and their comparison to the predefined objectives [Carrara et al., 1994]), or “creative” (including the fact-finding, ideas finding, and ideas evaluation [Osborn, 1963]), the creation and manipulation of design ideas remains the main common ingredient. The only exception is a type of design process when the “precedents” or “cases” are exploited as the starting points for design [Oxman et al., 1993; Flemming, 1994]. The cooperative design in the conceptual design stage can be found in [Gross, 1994].

Our work involving the distributed computer design agents started in a specific architectural domain where we investigated the design process utilizing an industrialized building system [Petrovic, Ibid.]. As we dealt with the “puzzle-making” type of design [Archea, 1987], or just “selections” [Alexander, 1968] from the field containing the complete set of possible designs, the point came when, in order to broaden the applicability of our design agents, a fresh approach was needed to introduce new agents, especially in the delicate stage of conceptual design,

To this end, we made an attempt to reproduce a typical “creative” design session, to see what new ideas we can get to help us define the performance criteria for the new design agents. As it happened, such opportunity came first in a non-architectural environment, in the School of Industrial Design at the University of Arts in Belgrade. We shall repeat the experiment with the architectural students in the autumn of this year.

To reconcile the need for a design context that is based on creation and proliferation of ideas, and the need to know what design agents to include and what kind of cooperation between them to expect in the process, we decided to analyze "brainstorming", a design technique that is dedicated to the proliferation of ideas by association by a group of people. The principles of brainstorming are well known, and can be found in the extensive literature on the subject [e.g. Osborn, Ibid.]. The primary emphasis in a brainstorming session is on creating a quantity of ideas on the selected subject by the participants' associations, while the evaluation of the ideas is performed later.

An Experiment: Simulation of Computer-Agents Brainstorming SessionTTTThe Sessions Scenarios

This brainstorming session was organized in a slightly different way than usual for the following reasons: 1. Various types of participants functional roles were introduced because we needed the identification of the behavior that these agents show in the association process, and 2. the format of the ideas the participants were supposed to create had to be pre-defined in accordance to the needs of our later work. Our assumption was that by studying the behaviors and the results of the participants work during the experiment, we would be able to define the expected performance of the computer design agents, and the type of their inter-communication. In the experiment, the human participants were expected to impersonate the future computer design agents.

Some characteristic types of the agents (reflected in their names) were taken from the traditional brainstorming session, some were borrowed from the list of the "mind agents" [Minsky, 1985]. The rest of the names were determined by the students themselves who also selected which agent they wanted to play. It was agreed that the students could exchange roles, or, all can play the same role if this was deemed necessary. The following agents were suggested: 1) Professor (a role by default, "leader", conducts the session, poses questions), 2) Associator (draws analogies, proposes concepts), 3) Builder (generates, "makes what can be made" - a Minsky's term), 4) Combiner (combines existing ideas into new ones - a Minsky's term.), 5) Concluder (makes generalizations), and 6) Coffee-girl (brings in new information). The ideas formats were borrowed from empirical psychology, cognitive sciences and neuroscience. In the first instance, these were: percepts, concepts, and prototypes, all "mental objects" [Changeaux, 1983],

The task of the experiment as a whole was to provide the initial data, and record the creation and transformation of the produced mental objects (which, in our context, became "design objects" when externally represented), and of the communications between the agents. The scenario of the first three sessions of the experiment included: transformation of percepts into concepts, creation of new concepts by associations, and transformations of concepts into prototypes. The last session was dedicated to the evaluation of prototypes into instances. The topic of the experiment was an industrial design product, but as we were interested in the process, and not the results as such, this fact was of no relevance for our investigation.

From percept to concepts

At the beginning of the first "creative" sessions, the Professor tried to obtain from the students as great a variety of percepts on the subject as possible. Here, the "characteristic words" from the brief were used to trigger-off the responses from the Associators, the role all participants except the Professor played at that time. The percepts were taken to be the hypotheses, not directly derived from the input information, but rather, of the internal models (in the participants' brains) selected on the basis of that information. Furthermore, the obtained percepts were not the isomorphic pictures of external shapes but crude continuos analogues of agents input-output functions in the presence of recognized objects. [after Gregory 1974].

The transformation of percepts into concepts is characterized by further simplifications.

"A concept, like an image, depends on memory, but its sensory component is almost non-existing. Loosing the "vividness" and isomorphy, it becomes completely abstract," ... however, ... "it is the associative characteristics of mental objects that enable them to be joined together, ... and make the chaining possible in an spontaneous and independent way" [Changeaux, Ibid.]. The task of the Professor was to unearth these characteristics and transfer them as possible links to the Associator(s), Combiner(s) and Builder(s) who were trying to propose the new ideas.

By the combinations of their traits and the formation of categories, concepts become "prototypes," the generalized objects that contain the main concept features. While the percept depends on the context stimuli, a prototype may have little or no connection to the context. A prototype can be memorized and retrieved using certain prototype key characteristics.

A prototype subjected to transformation creates new objects. When subjected to assessment of their relationships with a new context, prototypes produce instances, thus defining the “reality” of the design objects. All produced instances have some characteristics of the prototype but are molded into the different, respective contexts. Another way to appraise the validity of obtained design objects (percepts, concepts and prototypes) is to compare them with the initial data [Carrara et al., Ibid.]. This could cause the change of all objects due to the quite probable change of the initial context conditions.

Conclusions

Most participants followed the rules of the game, sometimes with an effort. The behaviors varied in performance. The Associators performed well, especially on the more general levels. When the issues became more specific, the importance of possession of specific knowledge was getting more important. We felt that a computer agent that can associate verbal clues with crude diagrams would be an important introductory link between the reality and the beginnings of design efforts, however feeble it shall be.

Associators, builders, and combiners of more complex concepts were indistinguishable in their roles most of the time. The logical thought was appreciated, but not always helpful. However, the students known, and often scolded for their “weird” ideas, “got their five minutes,” and produced the most innovative contributions. The Concluder (generalizer, categorizer) proved to be a very important role as only after he or she made sense of what had been produced in approximately every fifteen minutes, the continuation of a session could continue smoothly on.

The role of the Professor, a “meta-brain” of the brains taking part in the session, proved to be quite delicate. He was not only an inner “controller”, managing the pouring ideas of all kinds while maintaining the productive progress of the session, but also an implicit creator of ideas and a quick planner of future questions on the basis of the obtained ideas. He was taking decisions in real time on what questions to ask, to whom, and when to stop. He may had been helped by some prepared scripts, but had to use his instinct too.

The value of new information proved to be quite important. Although some of the participant stubbornly went “into depth” of some ideas, others often “reflected back”, not for evaluation of the proposed idea (which was forbidden), but for the support for getting into the divergence.

It is very important to stress that the participants communicated with each other the way they felt like, but mostly within their own terms of references. Minsky described a K-line as a “wirelike structure, that attaches itself to whichever mental agents are active when you solve a problem or have a good idea.” [Minsky, Ibid.]. In our experiments these lines emerged during the process, and were not repeated in the subsequent attempts.

A FRAMEWORK OF COMPUTER DESIGN AGENTS

Design Agents

Design process as a whole is comprised of design procedures, which, in turn, are composed of many small design processes performed by design agents. Each design agent is defined by the process function, the behavior it exhibits when performing the respective function, and the type of communication with other agents. It may have a local memory, or share the memory with other agents. The agents combine various product model representations, and processes of various complexity. Each agent produces a change of the design object related to the agent’s domain. Each state of a design object represents a temporary design “solution”.

Design agents ‘communicate’ with each other: an agent receives the input from other agents, acts on the design object producing the proposal (e.g. a local solutions to a problem), and sends the messages to other agents which can improve, modify, appraise or criticize the solution. The functionality of an agent does not depend on the functionalities of other agents. In case of the failure of the particular agent, the activity of other agents would not be affected, only the scope of the services offered by the framework will be reduced (e.g., some sort of design evaluation, or some representation of the evolving design shall not be available at that moment).

At present we distinguish the following generic classes of design agents: (a) builders or generators, (b) associators, (c) transformers (of pictorial into verbal representations and vice versa), (d) concluders, (e) combiners (f) evaluators, (g) form presenters (i.e., 2-d and 3-d modelers) and (h) others (i.e. managers and communicators). The particular members of these classes are engaged in the specific tasks (i.e., as creators of concepts, generators of form, transformers of concepts, generators of prototypes layouts, evaluators of room orientation, etc.). Different agents may be engaged in the similar tasks, and the specific tasks may be achieved by the different agents. For example, a rule-based

OYSTER may take an associator's function to make the associative interconnections between verbal and pictorial bases, accessing to images via verbal labels. [Pylyshin 1973]. This task can also be achieved by neural-net based agents, etc. The following list of agents is tentative, but necessary and sufficient to illustrate some meaningful design procedures. The list includes the agents previously developed for the GIMS-DDS system [Svetel, Ibid.]. Design of some new agents (e.g. Professor) is in development.

Examples of Developed Design Agents Prototypes

Viz-Verb (alias). Associator (generic name). A picture-verbal transformer for the conceptual level. Comprised of two feed-forward neural networks, each with a hidden layer. Together, they simulate the bi-directional association. Function: For given verbal input, Viz-Verb produces an associated pictorial symbol contained in the memory, and vice-versa, the verbal cues are produced for the given picture. The pictures may be drawn and edited in situ, or imported.

OYSTER-Combiner (alias). Combiner (generic name). OYSTER is a deductive inference shell with interchangeable knowledge-bases. Function: Combines concepts. Interaction with all agents which have the appropriate need for input/output of this type.

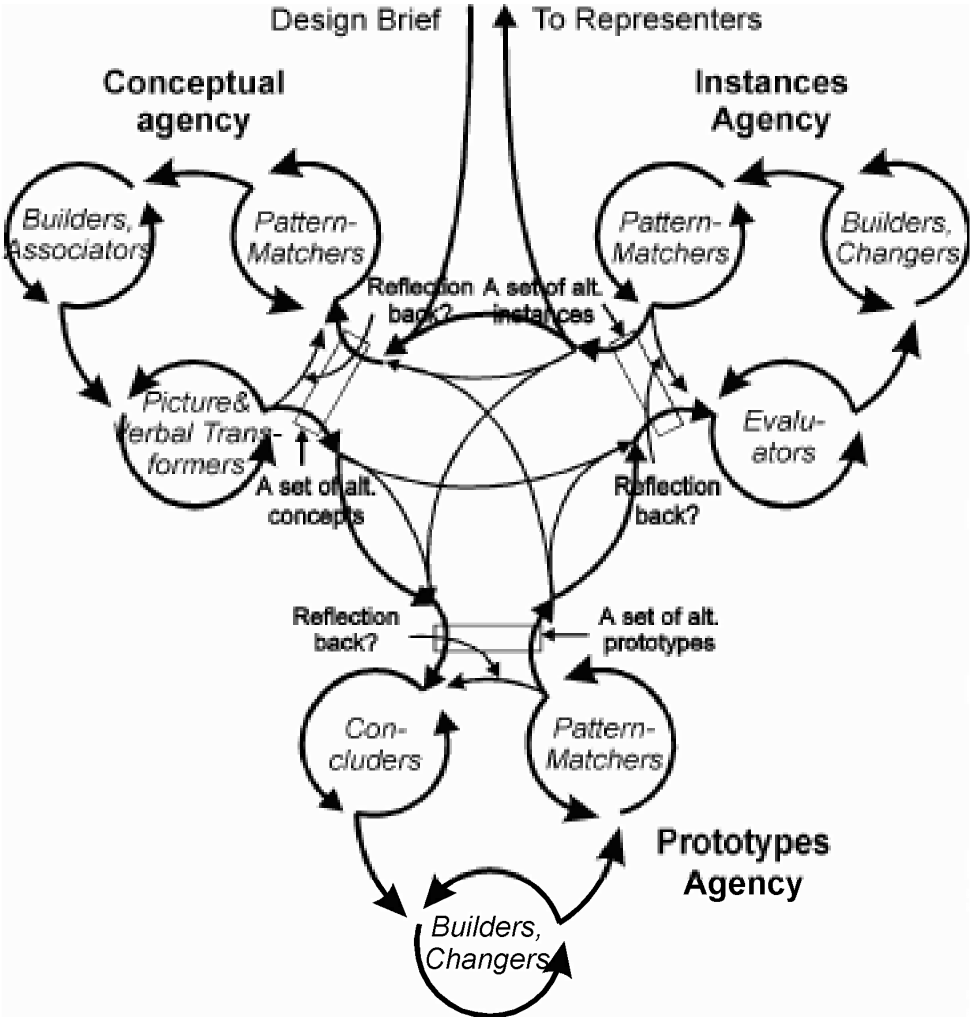
OYSTER-Evaluator-ONE, TWO, etc. (aliases). Evaluator (generic name). Evaluator's function: To make inferences related to the contents of knowledge-base. Interaction with all agents which have the appropriate need for input/output of this type. There exists several OYSTER knowledge-bases, such as evaluation of house orientation, evaluation of house character, etc.

ARCH (alias) Builder (generic name). The role: generator of GIMS industrialized system house layouts. Receives input information containing the required room-list, connections (between rooms) list, etc. During the action sends out the informing messages to other agents about the number of the completed designs. Output consists of files containing information on layouts list.

Layout Manager (alias). Data Manager (generic name). The role: data manager. Receives input information containing the layouts list. During the action sends out the requesting messages for consultation on orientation evaluation, subjective evaluation, request for making 3D representation of objects, etc. Screen output: house layouts.

Roof-Maker (alias) Builder (generic name). A rule-based system. The role: Generator of roof alternatives of a GIMS house. Receives input information: 3D building description

Figure 1.: A possible network of design agents (modeled after Hickling 1982, p. 284)



(without roof). Sends output information: alternative 3D building description (with roofs).

PDP-AAM-V.1.1. (alias) Associator (generic name). A 3-d Object-Verbal Transformer for the prototype level. It has the same structure as Viz-Verb, but the input/output layers are represented by a semantic differential and a 3-d editor, respectively. It behaves in such a way that when the input is presented by the object verbal description via semantic differential, on the basis of the memory base (the "trained" hidden layer), it would transform the input into a 3-d object representation using a 3-d editor. The process can be reversed: if an object 3-d form is given, the semantic differential is produced. Both Viz-Verb and PDP-AAM use the same mechanism based on the associative neural nets [Caudill et al. 1992] but obviously the other types of mechanism are also possible.

The design agents implementation environment is MS Windows 3.1 (Windows is a trademark of Microsoft Corporation), while the implementation language is Turbo Pascal for Windows 1.0 (Turbo Pascal is a trademark of Borland International Inc.).

Design Agents Communication

GIMS-DDS design agents “cooperate” when performing a procedure. Here, the cooperation is considered as the way to extend the scope of design expertise, taking into account various aspects of design. The basic communication between agents, a prerequisite for the cooperative, collaborative action, is represented by a set of the specific messages that correspond to the particular actions of the agents. Larger 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 stored in files. In addition to these communication protocols, a central blackboard is implemented to hold the global data that affect all agents in the system. All knowledge sources are realized as the separate files that establish the particular pieces of design competence. During the design session, new knowledge bases may be added to the various agents, but under the different names.

Communication of messages between agents is a condition of the primary importance for the cooperative design process. The second important issue refers to what is being communicated. At present, the files represent the model states. We can classify the following messages as types of ‘speech acts’ [Petrovic et al., 1994b]:

(1) *Directive one-way commands*: find, accept, reject, engage, select, do it, withdraw, abort, count, show, warning, stop, help, ...

(2) *Committing messages*: generate, build, add, subtract, magnify, (“evaluate” was not used in the described experiment sessions 1-3)

(3) *Declarative messages*: Facts lists about the object, ...

(4) *Assertive messages*: “This object corresponds to the brief,” ...

(5) *Expressive messages*: ‘The user dislikes this type of object,” ...

(6) *Inquiring messages*: “What new information is available?”, “Why did (not) you do that?” ...

Table 1: Types of computer agents messages [after Winograd and Flores, 1985. p 59]

Figure 2.: The performance of the ad-hoc design agents agency

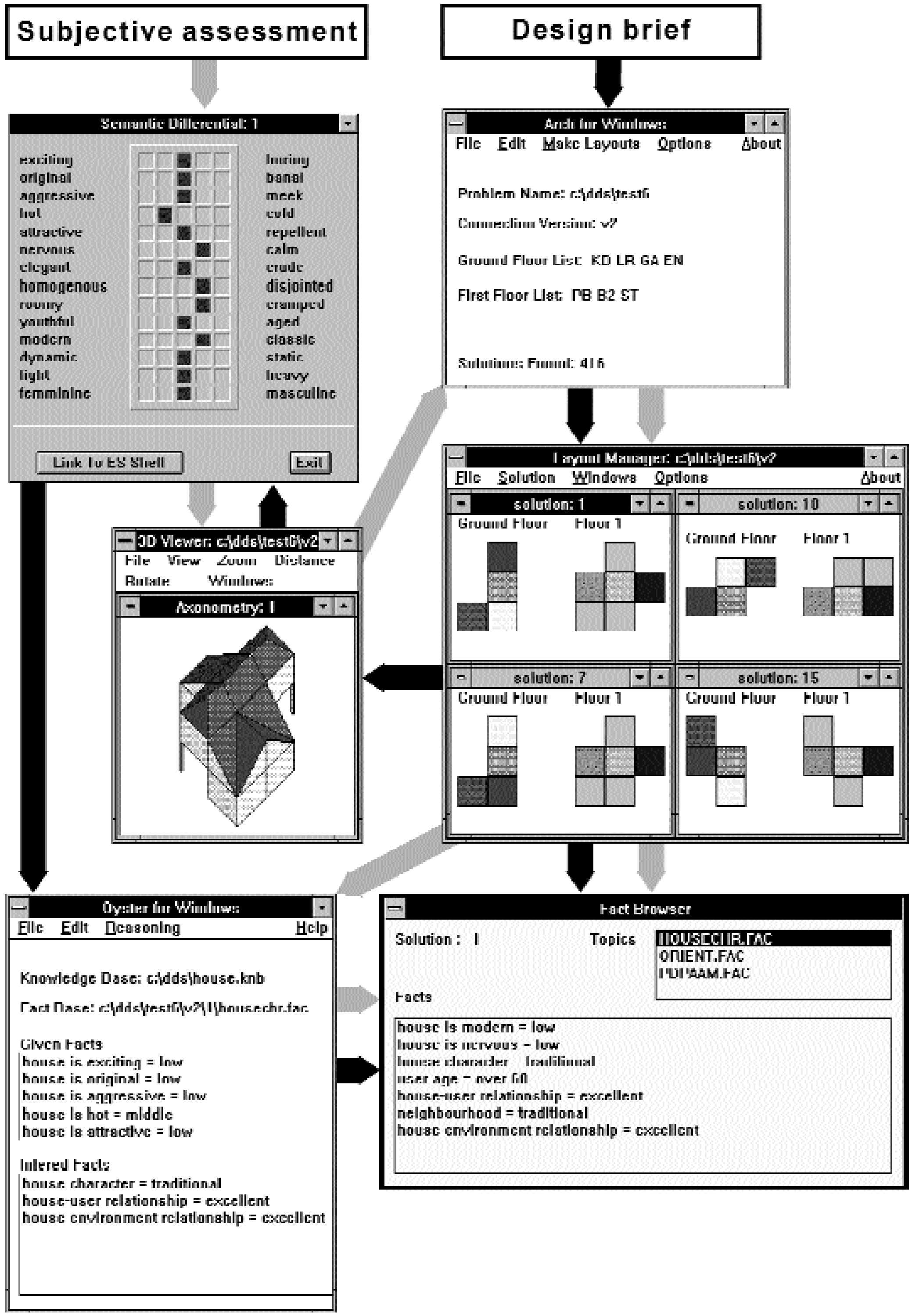
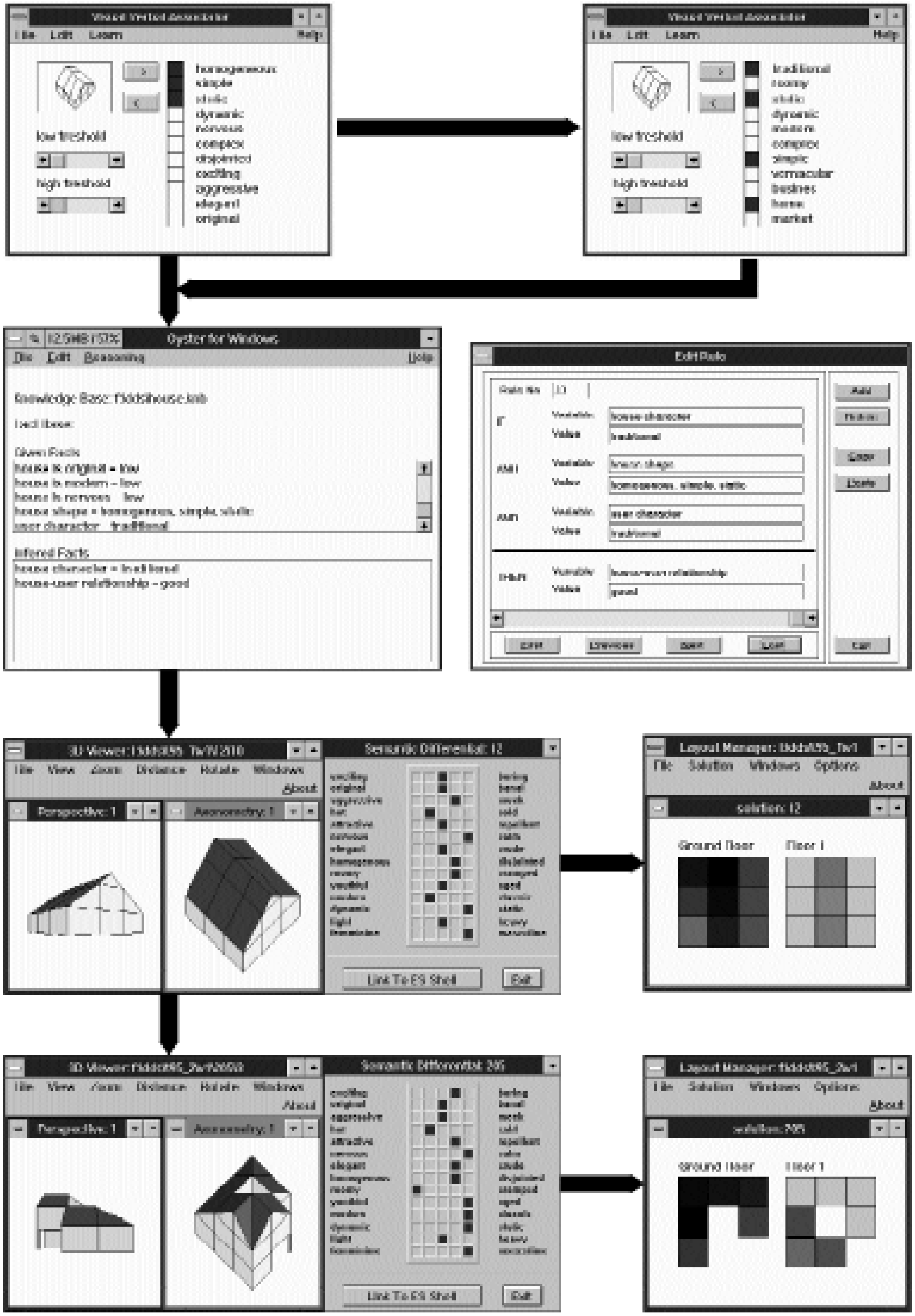


Figure 3.: A design agents agency allowing the “traditional” design path



Design Agencies

Design agencies perform design procedures that consist of design processes performed by the individual design agents. A design agency is any assembly of design agents considered in term what it can accomplish as a unit, without regard to what each of its parts does it by itself [after Minsky, Ibid.]. An agency is comprised of at least two design agents. It is a smallest "design agents society" that can produce the meaningful results. An agency can be described in terms similar to those used for the description of design agents. Each design agency may have a memory shared by its agents. The particular traits of some agencies may also be in their concurrency, recurrency and reversibility. Agencies may be assembled ad-hoc, for the specific design task, or be permanent. The organization of a design agency may be prompted by an inner or the outside agent - the Human Designer, a Meta-Agent - a member of a larger Agency, or may emerge during the design process.

The Framework

The agents relationships determine the circular, "whirling" type network [Hickling, 1982] presented on Figure 1. Within this structure, in principle, the design process can start at any point, and go from there to any other point. If a starting point has no new input data, (i.e., the 2-d or 3-d Viewer), there is always a library of precedents that may serve as a starting product model. The process may also finish at any point of the network.

DESIGN AGENCIES AT WORK

Some Possible Paths

Let us assemble an ad-hoc agency comprised of the following agents: Viz-Verb, OYSTER, PDP-AAM, ARCH and Layout Manager (Figure 2). Each agent has a two-way relationship with each other agent thus allowing an agency to execute a variety of paths as presented in Figure 1.

We shall start with a partial input coming from the design brief which is often includes some characteristic cue-words describing the desired object. For example, a "solution" might be expected to exhibit such traits as: simplicity, modesty, traditional look (blending with the environment) and the like, or may be presented by a crude sketch. For our purposes, we can consider these verbal and pictorial representations of an object, however construed, as investor's or designer's percepts (or concepts). The first thing we can do is to make the transformation of a verbal representation into a pictorial one, or vice-versa. For this we would have to use Viz-Verb as many times as necessary, depending on the key-words and the corresponding memory-bases. The Figure 2 presents two such occasions where the first application was supposed to interpret the shape of a house, the second, a house considered as a dwelling. Once we had a concept representation of an object, we could proceed with making combinations of concepts, and their associations. Our second agent, OYSTER, can do that, but it can also interpret all verbal inputs and produce a series of more complex verbal strings that define a "semantic differential" of an object. Such, more flexible and potent verbal structure may lead to making of prototypes, the next important task of our agency.

For a defined semantic differential, PDP-AAM shall produce the prototype alternatives. Then, we can employ "ARCH" and "2-d Viewer", to generate and illustrate the alternatives of functional spaces and their interrelationships that could be realized in the produced building. Finally, OYSTER could conclude whether so defined house description agrees with some other requirements in the design brief (e.g. agreement with the physical context, the users age, housing regulations, etc.).

We have now described only one script of a design object transformation. However simplified it may be, we believe that such experimental agencies may be used in an educational environment, leading to better understanding of principles of knowledge-bases, and definition of concepts. On the other hand, experimenting with PDP-AAM and the very efficient ARCH enables the investigation of the relationships between the shape and the content of a house in various contexts. Many more paths are possible. For example, if the newly produced outputs by neural nets are learned, each consecutive attempt by the agents shall produce the different symbols/prototypes. Furthermore, an outside user may self-define all key-words and the semantic differential, train the neural nets with different examples, and use his/her own evaluative knowledge-base with OYSTER. The same inputs shall now produce completely different outputs. Each obtained output by Viz-Verb or PDP-AAM can be "reflected-back".

The Figure 3. shows some more possibilities. The first path, starts with a vague description of a desired house, expressed by any interested party. The learned examples can represent anyone's subjective (or /and objective) preferences. The PDP-AAM would produce a house prototype, that was most probably not learned, and give the task to ARCH to see what internal arrangements can be made in the house shell. OYSTER would make some conclusions about, say, house fitting into the environment. Another possibility would be to start the design process in a "normal" way, from the list of required rooms, then creating the layouts, make 3-d objects, and each produced prototype “subjectively evaluate” by PDP-AAM. Or, one can start with any of the existing agents, and see to what results this exploration would lead.

Conclusions

In view of our earlier work on development of various design agents and their limited applications in design studio and school environments, we believe that by using our framework, the students (or designers) may get an insight into the perplexity of design process, and assess the feasibility of computer-assistance in not only the objective but also in the subjective evaluation of design objects in course of design process. We have presented the agents to two architectural schools and few offices, and, after the initial shock of dealing with so simple houses and crude knowledge-bases has passed, and after our assurance that we were not making the “artificial designers” but helping the human designers to use their minds more fully, the ideas started to come. In offices, much valued feature of our agents was the generation of the very different urban tissue elements with so few basic forms.

So far we have developed the prototypes of some of the more important design agents to prove that the idea works, mut much work is ahead. We are fully aware that we have just scratched the surface of the problems, especially in view of the time scale, engaged resources and the current state of our knowledge. Some of the agents like Viz-Verb and PDP-AAM are strange creatures: due to their nature that escapes formalization, no explanation of their is possible. Moreover, if given the objects descriptions that are not contained in the memory, they would make the generalizations of what had been learned, and bravely produce the “novel” pictorial/verbal descriptions.

One of the features of the described framework is that many agents developed elsewhere, if fitting into the software environment, could join the present society with the minimum effort of making the communication links. In the next phase of the project we shall try to prepare more worked-out examples to be able to assess tyhe pros and contras of our work, and make the strategic decisions on the project future.

Acknowledgements

Obviously, our work draws heavily on the ideas of many people to whom we owe our thanks. Thanks are also due to Igor Svetel, for not only the technical, but also the moral support in writing this paper. He has developed all computer design agents that appear in the paper. The responsibility for the use of the borrowed concepts in the new context and for the soundness of the theses presented remains entirely on the present author. Acknowledgments are also due to staff and the 3rd Year students at the Industrial Design Department of the Faculty of Applied Arts and Design in Belgrade who took part in the experiments during the Spring Semester 1995. The work described in the paper is a part of the project financed by the Institute IMS and the Ministry of Science and Technology of the Republic of Serbia. The project director is Ivan Petrovic.

References

Alexander, C., Ishikawa, S., Silverstein, M., A Pattern Language Which Generates Multi-Service Centers, Berkeley, California: Center for Enviromental Structure, 1968.

Archea, J., "Puzzle making: What architects do when no one is looking," The Computability of Design, edited by Y.E., Kalay, New York: John Wiley and Sons, 37-52, 1987.

Caudill, M., and C., Butler, Understanding Neural Networks: Computer Explorations, Vols I and II. Cambridge, Massachusetts and London, England: MIT Press, 1993.

Changeux, Jean-Pierre, L'Homme Neuronal, Paris: Fayard, 1983.

Carrara, G., Kalay, Y.E., and G. Novembri, "Knowledge-based computational support for architectural design", Knowledge-Based Computer-Aided Architectural Design, edited by G. Carrara, and Y. Kalay, Amsterdam: Elsevier, 141-201, 1994.

Coyne, R., "Objectivity and the Design Process," Environment and Planning B, 18.3: 361-372, 1991.

Coyne, R. D., "Tools for Exploring Associative Reasoning in Design", The Electronic Design Studio, edited by M., McCullough, W.J., Mitchell, and P. Purcell, Cambridge, MA, London, England: The MIT Press, 91-106, 1990.

Flemming, U., "Case-based design in the SEED system," Knowledge-Based Computer-Aided Architectural Design, edited by G. Carrara, and Y. Kalay, Amsterdam: Elsevier, 69-91, 1994.

Gregory R.L., Concepts and mechanisms of perception, London: Duckworth, 1974.

Lakoff, G., Women, Fire and Dangerous Things, Chicago, Il: University of Chicago Press, 1987.

Minsky, M., The Society of Mind, New York: Simon & Schuster Inc., 1985.

Osborn A., F., Applied Imagination, New York: Charles Scribner's Sons, 1953, Third Revised Edition, 1963.

Oxman, R., and R. Oxman, "PRECEDENTS: Memory Structure in Design Case Libraries," CAAD Futures '93, Proceedings of the Fifth International Conference on Computer-Aided Architectural Design Futures, edited by U. Flemming and S. Van Wyk, Amsterdam: North Holland, 273-288, 1993.

Petrovic, I., "On Some Issues of Development of Computer-Based Architectural Design Systems," Knowledge-Based Computer-Aided Architectural Design, edited by G. Carrara, and Y. Kalay, Amsterdam: Elsevier, 269-301, 1994.

Petrovic, I., and I., Svetel, "Conversation for Design Action: by Men or by Machines?" The Virtual Studio, Proceedings of the 12th Europeean Conference on Education in Computer Aided Architectural Design, September 7-10, 1994, edited by T., Maver and J., Petric, Glasgow, Scotland: The Strathclyde University, 37-47, 1994.

Petrovic, I. and I. Svetel, "On Some Aspects of Product Modelling in Architectural Design: GIMS-DDS Design System". CIB W78 Workshop On Computer Integrated Construction, Helsinki, Finland, Aug. 22-24, 1994.

Pohl, J., and L., Myer, "A distributed cooperative model for architectural design:, Knowledge-Based Computer-Aided Architectural Design, edited by G. Carrara, and Y. Kalay, Amsterdam: Elsevier, 205-242, 1994.

Pylyshin, Z.W., "What the mind's eye tells the mind's brain: A critique of mental imagery," Psychological Bulletin 80: 1-24, 1973.

Rutherford, J.H., and T. W., Maver, "Knowledge-based design support", Knowledge-Based Computer-Aided Architectural Design, edited by G. Carrara, and Y. Kalay, Amsterdam: Elsevier, 243-267, 1994.

Svetel, I., "DDS - Distributed Design System: A Paradigm for Modeling Conceptual Architectural Design," Advances in Computer-Based Design Systems, edited be J., Pohl, Baden-Baden: The International Institute for Advanced Studies in Systems Research and Cybernetics, 79-88, 1994.

Schˆn, D. A., "Designing as reflective conversation with the materials of a design situation," Knowledge-Based Systems, 5.1: 3-14, 1992.

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