

Integration of reasoning systems in architectural modeling activities

Halime Demirkan

Faculty of Art, Design and Architecture, Bilkent University, 06533, Bilkent, Ankara, Turkey

Abstract

In the first phase of a design process, the designer understands the problem and assimilates it to a conceptual framework that is already known to him. Due to the nature of design problems, the reasoning methods and techniques for modeling are not uniform and clear. An integrated reasoning system is proposed for modeling the architectural processes. Such a system may help designers to make decisions based on past experiences as well as domain theory. The performance of the integration approach is compared with the pure case-based and rule-based reasoning systems to study the efficiency and effectiveness in the same domains. The study tried to identify the reasoning systems used by designers pertaining to the interior design applications. © 1998 Published by Elsevier Science B.V.

Keywords: Integration of reasoning systems; Rule-based reasoning; Case-based reasoning; Architectural design process

1. Introduction

The architectural design process distinguishes from other problem solving tasks in many ways. The first characteristic of architectural design processes is the generation of new states from the current ones. The states representing the candidate solution are generated before they are evaluated. The second characteristic that guides the search relies not only on the information internal to the particular problem, but also on the information that is external to it, such as cultural norms or style. The designer uses reasoning methods in all these states. The reasoning methods that are used in design are not always fully understood. Although the literature abounds in methods and techniques for modeling reasoning process, their description is not uniform and unambiguous. Knowledge of principles and methods may be stored in the memory of the designer, but these should be interpreted by the designer.

Usually knowledge is thought to be something that we keep in our minds. As Frost [1] states, “it is stored in the neural system of human beings, possibly in the networks of nerves involving synaptic connections of variable electro-chemical conductive which is a kind of soft-wired circuit whose structure and function can be modified through experience” (p. 1). Yet, its exact physiology is unknown.

Purcell and Gero [2] claim that, “designing can be viewed as an activity in which expert knowledge is used to fill out an initial statement of problem so that a solution can be developed” (p. 82). Initially the designer should obtain knowledge about what is to be designed and the constraints and requirements of the designed artifact. Also, the designer should use the design knowledge that covers a broad spectrum, including laws, rules and formulas pertaining to the behavior of people, materials, objects and spaces. The knowledge with which we are concerned in design may apply to different abstractions of the

design process and to different abstractions of the design description. There is knowledge about how components might fit together in a design. There is also knowledge about appropriate actions to perform in producing configurations and knowledge about strategies. We must be able to represent and manipulate knowledge of this kind.

There are two ways of representing the design knowledge, namely, the representation of the artifacts and the representation of the design processes. Progress has been made in the representation of artifacts. However, almost no significant results have been seen in the representation of design processes, which implies that a design model is needed [3].

2. Reasoning modes in modeling activities

A designer does not use algorithms to solve problems. In an algorithm a goal is assumed and a series of steps carried out in a logical order to lead to that goal. Designers solve problems that are less well defined and the methods they use are not fully understood [4]. In solving design problems, the extent of experience of the human being is very important. It consists of much more than just facts and rules. Designers while practicing their professions develop rules of thumb or heuristics. Unfortunately, designers are unaware of the details of how they work. They may use assumptions or beliefs without being explicitly aware of them. The ability to explain an answer or line of reasoning is a necessary feature of a designer. Goldschmidt [5] investigated the reasoning processes supported by sketching. The analysis focused on the verbal transcripts of utterances produced by designers. Maziloglou et al. [6], claim that these studies did not provide a detailed analysis of workspace activity nor attempted to relate ideas to design workspace behavior.

There are three modes of scientific reasoning: deduction, induction and abduction. Deduction is reasoning from the general to the particular. In deduction we are given a rule from which it is possible to deduce facts about specific cases. In induction given a set of specific examples, we investigate the examples and induce rules and patterns. Induction is about generalizing from individual instances to some

hypothesis (as an idea, a theory or a model), and abduction is the converse: producing a thing that belongs to a particular case.

Deductive reasoning mode is well defined and is the basic building block of formal reasoning systems. In science, induction is regarded as the process by which theories are derived from observations of certain phenomena. The third mode of reasoning, abduction, has been proposed by Pierce (1839–1914) in Feibleman [7]. Abduction is the derivation of statements about the world given logical rules and some logical consequences. Problems arise when one reasons deductively with abductive rules. The crucial point is the deductive knowledge from which the abductive rules are derived should represent a closed-world—that is a world in which no knowledge exists other than stated [8].

Adopting the model of reasoning, there are then three tasks connected with design. The first is the creation of an artifact that is accomplished by abduction. Second, the prediction of performance characteristics of this design is accomplished by deduction. Given the characteristics of the design and certain laws, the behavior of the design can be predicted (deduced). Third, the accumulation of habitual notions, established values and evolving typology (design descriptions) are accomplished by induction.

The designer is subjective, forgetful, omits details and may be inconsistent. He finds it easier to refer to specific examples than to describe his process. If he can provide a set of examples of cases, consisting of his decisions together with rules, facts (such as legislation or codes) which he considered in making those decisions, then he can induce rules from those examples. Integrated reasoning system is a framework of design process and explains how design conceptually is performed in terms of knowledge manipulation.

3. Reasoning systems in modeling activities

There are many different ways in which knowledge can be represented by designers; it is a desirable thing to represent this knowledge in a uniform way. A case-based system is appropriate for an experience-rich domain, while a rule-based system performs reasonably well in a knowledge-rich appli-

cation environment [9]. One class of these systems, the production reasoning systems, is concerned with transforming some descriptions of the world by means of rules. The transformation process involves both deductive, inductive and abductive reasoning agents. In production rule-based systems, designer determines which rule will be executed. The design knowledge may be expressed as set of transformational rules in which an antecedent is replaced by a consequent.

While a designer uses his knowledge, he uses both the observed and the derived facts that are deduced through a justified mode of inference. Knowing about the world, one can provide derivable facts. The way of applying rules, which are the modes of reasoning, differs from case to case. The human being as an interpreter defines how a mode of reasoning should be used to achieve the desired ends. As an interpreter he sets the goals and objectives. Sometimes priorities are applied in order to select between competing knowledge sources [10].

Another class is the case-based reasoning and it is useful when knowledge is incomplete or evidence is sparse. It is defined as reasoning from old cases or experiences in an effort to solve problems, critique solutions, explain anomalous situations or interpret them [11]. The designer may be influenced by the previous designs of himself or others in design cases. A designer creates the artifacts by remembering similar previous designs and comparing and contrasting the new one to the old ones. Designers use their own experiences if they have a relevant one, or they make use of the experience of the others to the extent that they can obtain information about such experiences. An individual's knowledge is the collection of experiences that he has had or that he has heard about or seen.

In case-based reasoning, a designer solves new problems by remembering previous situations similar to the new situation. Case-based reasoning in design can mean adapting old design solutions to solve new design problems, using old cases to critique new solutions, or using old cases to explain new design solutions. The designer's memory depends upon good methods of labelling cases, so that they can be retrieved when needed. General principles such as rules or patterns that he abstracts from a given case or set of cases help him label cases for retrieval,

determine which cases are relevant and adapt those parts of cases that do not fit the current design situation [11].

A designer uses several design cases to solve complex design problems. In design, the problems are too large to solve as one chunk and need to be solved in parts, but the parts interact with each other in strong ways. While using multiple cases, the designer checks the consistency of the proposed solutions [10]. Although cases can suggest solutions, the designer needs to provide processing control for the consistency between these solutions. Also, the designer responds to changes in the world and adapt the solutions to the existing situations. The designers investigate the reuse of old plans in a dynamically changing world.

There are certain issues to be analyzed for better understanding of case-based reasoning in design as follows [11].

How experience can be represented and organized in design solutions?

At what points in reasoning is design experience used? How design experience is used at these points? What role does design experience play in reasoning?

What kinds of performance differences can we expect in a novice designer compared to an expert designer?

How does the evolution from novice to expert designer happen?

What retrieval strategies are necessary to access appropriate previous design experiences?

Certainly, designers are happy to tell the rules that they use, but whether the designers actually use such rules when they reason is another question. There is a difference between textbook knowledge and actual experience. Most designers will tell us about their experiences, but the real question is how that experience is encoded in their memories. In design where the problem is not clear, the designers frequently cite previous cases that they have worked on that the current case reminds of them. Obviously, in case-based reasoning the designer uses rules. When a design activity has been repeated often enough, it becomes rule-like in nature.

When a rule fails, the only alternative for the designer is to create a case that captures that failure. A new rule is created if those design cases are sufficiently alike. Using abduction the designer de-

rives new rules. If the cases from which they were originally derived are unknown and they are stored by large numbers of people, they are called heuristics. However, the real world consists of situations that are unique and sometimes even contradictory. If the designer has only had one experience that was in any way relevant to present situation, he will reason using a paradigm. By reasoning the designer will find where the current design differs from the paradigmatic case and adapt it to help him in the new design case.

4. Modeling activities

Knowledge acquisition is a different process in design and yet, there are no formal methodologies that have proved to be effective. As a general principle, the designer should be encouraged to describe his expertise in the way which is most natural to him. According to the studies in this field [12–16], it is recognized that rule based-reasoning and case-based reasoning are used by designers.

When we analyze the design process, we see that the first phase consists of understanding the problem, that is knowledge acquisition. In practice this often means understanding the problem and assimilating it to a conceptual framework that is already known to the designer. Barbuceanu [17] stated that modeling is the intellectual, creative component and, moreover, it is essential for obtaining a correct solution in the first place. He also adds, “the existing methods and tools have been developed almost exclusively for the programming side, while modeling is generally left entirely on the shoulders of the human” (p. 246). The goal of the author of this article is to construct a new brand of knowledge environment that would support the range of knowledge in modeling activities.

Knowledge acquisition is an active modeling process. A designer constructs a conceptual model of the artifact by abstracting knowledge from previous experiences and information stored in the memory. This abstraction process is aided by the use of interpretation. For example, a *kitchen* invokes responses of such spaces that already exists in the designer’s mind. The structures are used as templates

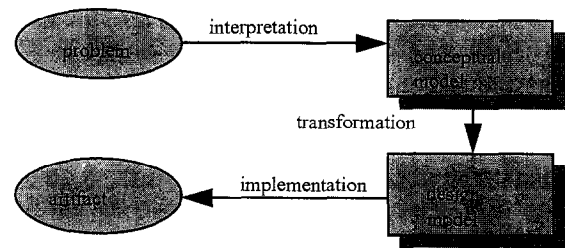


Fig. 1. The design process.

to understand program requirements and to guide the problem solving strategy. The designer determines his own priorities on the acquired knowledge. He recalls his analogies, memories and forms the pre-solution model. These conceptual representations are being linked both with the external forms of knowledge and with the internal representations of the model.

The conceptualization of the problem space starts with the existence of a problem that needs to be solved. Needs and desires are stated by a client and the design solution should satisfy him. The output of the design process is a design solution and not an artifact. As an example, the design of a kitchen needs to be realized in the actual kitchen, but this is not considered as a part of the conceptual model.

The description of the representational techniques that the artifact is realized, is the design model. The design model describes how the conceptual model is realized with representational techniques that are the various drawings or computer representations of the artifacts. Fig. 1 shows the dependencies of the models discussed in this section. Connections indicate that information from one model is used in the construction of another model.

4.1. Interpretation

Cross and Cross [18] state that analyzing and understanding the problem is an inevitable part of the design problem. In order to analyze the problem, information relevant to the task has to be gathered. In addition, relevant information has to be extracted from its source and shared with the design team. Some errors may occur in understanding the design requirements. The authors point out that there may

be misinterpretations of the information and some requirement may be forgotten.

Both knowledge and design policies are stored in the memory of the designer. Knowledge may be the truth or rule of thumb that usually does not change over time, while the latter can be modified according to the designer. Therefore, an old design may be treated in different ways on the basis of the designers or at different times by the same designer.

Knowledge comes from different sources as seen in Fig. 2. These involve knowledge from media as books, journals and videotapes. Also, the observed cases are another source. These are the previous experiences of the designer or others. The last source is the requirements stated by the requirers, experts or clients [19]. All the knowledge obtained from experts should be transferred to a domain without distortion.

The *artifact description* component maintains the current description of the design object. As an interpretation of the needs and desires, a requirement description is constructed in the designer's mind (Fig. 3). For example, a customer may state the need for a dining kitchen for a family of four. The designer stores the information in his memory as a kitchen. Design descriptions are the artifacts in the knowledge domain of the designer. They can have properties and relations associated with them.

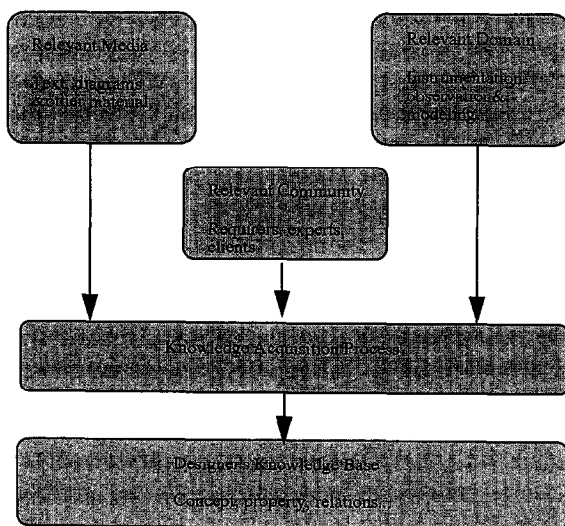


Fig. 2. The knowledge acquisition process [19].

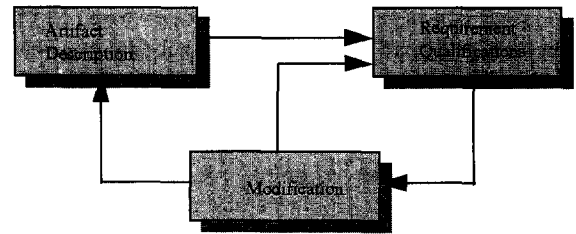


Fig. 3. Interpretation of the designer.

Every artifact has descriptive and functional knowledge associated with it. The descriptive knowledge is made up of properties of the artifact such as form geometric location. Properties are defined through their names and the description of the values that the property can take. As an example, the base unit in a kitchen with sink in it has a certain form and its geometric location should be related to the water source.

The functional knowledge describes how the artifact should be manipulated and used, and what the relationships are between various parameters. There are relations between the design artifacts. These relations may be stated as the subclass relation or the part-of relation. A second type of relation is the relation between expressions about property values. Part-whole relationships link objects in a hierarchical structure. A sink, for example, is a part of a base unit, which is a part of a kitchen. If a sink is going to be eliminated, since it is a part of a base unit, it should be replaced with a base unit in order to have the continuity of counters.

The *requirement component* contains the set of requirements of the artifact obtained at the end of the design process. As an example, the client may require a dining kitchen for a family of four, but the requirements for a household of four should be expanded by the designers as:

- total area of a dining kitchen has to be 15 m²,
- area of eating surface has to be 4.6 m² with a length of 1.2 m and width of 2.44 m,
- total cupboard frontal length has to be 5.9 m,
- total shelf area has to be 6 m², and
- others.

The designer as a *modifier* maintains knowledge about the design artifacts. The knowledge about the

design objects is used to deduce (draw) conclusions from the information already stored in the memory of the designer. This component also is used to check whether the current design description is consistent with the domain knowledge or not, if there is inconsistency this has to be recorded.

The designer as a modifier has knowledge about how to make the necessary changes. The design object may be decomposed into more specific components in order to be modified. The designer stores in his memory the information about the changes according to the essential qualifications.

4.2. Conceptual model

The conceptualization of the problem space starts with the existence of a problem that needs to be solved. As Foz [14] stated, we are limited to the approach that design only consists of logical manipulations of the abstracted properties of artifacts. There are other aspects of design that are unclear even to the designer himself. There is a general belief that the problem solving in design proceeds best by reasoned stages from the fundamental to the sophisticated. Foz [14], in his observations, found out that designers do not really work from elemental to the complex as thought. The general problem solving approach was first to find a pre-solution model drawn from their memories; second is to apply it as a template to the problem and work out the details. This is depicted in Fig. 1.

Conceptual model is the representation of inferential knowledge. Akin [12] describes knowledge-model based on the behaviors of human subjects as: design-symbol representations, transformation rules and heuristic rules. Also, Akin [13] points out that conceptual inferences differ from logical inferences in terms of spontaneity, probabilistic nature and chances of causing errors and functional utility.

Brazier et al. [20] stated that the view of design as a complex reasoning process has gained considerable influence with taking place at various levels of design and involving different types of reasoning, since the second half of the eighties. They classify the formation of conceptual model into two aspects, namely; static and dynamic. Static aspects not only include domain knowledge about properties of de-

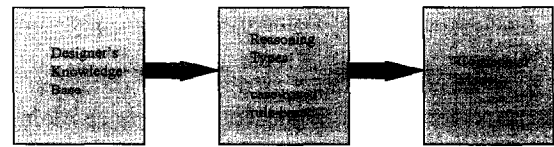


Fig. 4. Reasoning in formation of conceptual models.

sign artifacts and relations between these properties, but also domain knowledge about requirements of artifacts and relations between these requirements. Dynamic aspects include strategic knowledge about steps undertaken in the design process. To describe the dynamics of design process the circumstances under which a choice among these alternatives is made, must be specified. The reasoning modes involved in the process should be identified (Fig. 4).

Wielanga et al. [21] describe conceptual models as, “abstract descriptions of the objects and operations that a system should know about, formulated in such a way that capture the intuitions that humans have of this behavior” (p. 12).

4.3. Transformation to design model

The model uses reasoning to select and transform specific solutions to design problems to be appropriate as solutions for a new design problem. An integrated reasoning system unites a case-based reasoning system and a rule based reasoning system that applies domain theories to perform problem solving tasks. This system eliminates some of the drawbacks of pure inductive reasoning methods and pure deductive reasoning approaches. The domain theories that include both domain knowledge and policies are represented as rules. The designers can enter facts or encode design policies as generalization rules in the domain theory.

The aim of a design model is to construct an artifact. There are a lot of different requirements that may be contradictory to each other. Also, there is usually more than one solution that meets the requirements to a certain extent. A design solution cannot be derived from the requirement by a client. Simon [22] presents design as a problem solving process, and, even more specifically, as a search process. Due to the characteristics of design, using search as model for the design process is too general.

Maier [23] identifies three distinct models of design naming as: ‘decomposition, case-based reasoning and transformation’ (p. 51). In decomposition, large complex problems are divided into smaller, less complex problems. In a case-based reasoning model, analogical reasoning is used to select and transform specific solutions of previous design problems to be appropriate as solutions for a new design problem. In the transformation model, the design knowledge is expressed as a set of transformational rules in which the left-hand side of the rule is replaced by the right-hand side of the rule.

Chandrasekaran [24] proposes a task structure for design by analyzing a general class of methods called propose–critique–modify (PCM) methods. Chandrasekaran groups design proposal methods into three classes as: (1) problem decomposition–solution composition, (2) retrieval of cases from memory, and (3) constraint satisfaction. The methods are declared to have “the sub tasks of proposing partial or complete design solutions, verifying proposed solutions, critiquing the proposals by identifying causes of failure if any, and modifying proposals to satisfy design proposals” (p. 62). The design model is phrased in the terminology of the artifact.

The author claims that decomposition by Maier [23] or problem decomposition–solution composition proposed by Chandrasekaran [24] is only techniques used in design process. Constraint satisfaction proposed by Chandrasekaran is an operations research technique. These techniques are used in modeling and cannot be considered as models. Therefore, the reasoning is the most important issue in modeling and integration of reasoning systems has to be considered as the milestones in modeling.

5. Conclusion

In this paper, an integrated reasoning system has been proposed for modeling architectural process. Such a system may help the designers to make decisions based on past experiences as well as domain theory. The performance of the integration approach should be compared with the pure case-based reasoning systems and the pure rule-based-reasoning systems to study the efficiency and effective-

ness in the same domains. The current implementation of the integrated reasoning system is limited to interior design applications.

References

- [1] R. Frost, Introduction to Knowledge Base Systems, Macmillan, New York, 1986.
- [2] A.T. Purcell, J.S. Gero, Effects of examples on the results of a design activity, *Knowledge-Based Syst.* 5 (1) (1992) 82–91.
- [3] H. Takeda, P. Veerkamp, T. Tomoyama, H. Yoshikawa, Modeling design processes, *AI Mag.* 11 (4) (1990) 37–48.
- [4] A. Hart, Knowledge elicitation: issues and methods, *Comput.-Aided Design* 17 (9) (1985) 455–462.
- [5] G. Goldschmidt, The dialectics of sketching, *Creativity Res. J.* 4 (2) (1991) 123–143.
- [6] M. Maziloglou, S.A.R. Scrivener, S.M. Clark, Representing design workspace activity, *Analysing Design Activity in The Delft Protocols Workshop*, Delft, 1994, pp. 32–53.
- [7] J. Feibleman, An Introduction to the Philosophy of Charles S. Peirce, Cambridge, MIT Press, MA, 1970.
- [8] R.D. Coyne, Design reasoning without explanations, *AI Mag.* 11 (4) (1990) 72–80.
- [9] R.T.H. Chi, M.Y. Kiang, Reasoning by coordination: and integration of case-based and rule based-reasoning systems, *Knowledge-Based Syst.* 6 (2) (1993) 103–113.
- [10] H. Demirkan, M. Pultar, B. Özgüç, A knowledge-based space planning system, *Architectural Sci. Rev.* 35 (1) (1992) 3–7.
- [11] J.L. Kolodner, *Cased-Based Learning*, Kluwer, Dordrecht, 1993, pp. 1–5.
- [12] Ö. Akin, AIM: Architectural inference maker, CAD-78, Proceedings of the Computer Aided Design Conference, Sussex, England, 1978, pp. 506–519.
- [13] Ö. Akin, How do architects design? in: J.C. Latombe (Ed.), *Artificial Intelligence and Pattern Recognition in Computer-Aided Design*, North-Holland, New York, 1978, pp. 65–98.
- [14] A. Foz, Observations on designer behavior in the parti, Proceedings of the Design Activity Conference, London, 1973.
- [15] U. Fleming, Knowledge representation and acquisition in the LOOS system, *Building Environ.* 25 (3) (1990) 209–219.
- [16] J.L. Kolodner, Improving human decision making through case-based decision aiding, *AI Mag.* 12 (2) (1991) 52–68.
- [17] M. Barbuceanu, Models: toward integrated knowledge modeling environments, *Knowledge Acquisition* 5 (1993) 245–304.
- [18] N. Cross, A.C. Cross, Observations on teamwork and the social process of design, *Analysing Design Activity in The Delft Protocols Workshop*, Delft, 1994, pp. 1–18.
- [19] A.T. Rapoport, B.R. Gaines, Integrated knowledge base building environments, *Knowledge Acquisition* 2 (1990) 51–71.
- [20] F.M.T. Brazier, P.H.G. Van Langen, Zs. Ruttkay, J. Treur,

- On formal specification of design tasks, in: J.S. Gero, F. Sudweeks (Eds.), *Proceedings of the Artificial Intelligence in Design' 94*, Switzerland, 1994, pp. 535–552.
- [21] B.J. Wielanga, M.A. Schreiber, J.A. Breuker, KADS: a modelling approach to knowledge engineering, *Knowledge Acquisition* 4 (1992) 5–53.
- [22] H. Simon, *Sciences of the Artificial*, MIT Press, Cambridge, 1969.
- [23] M.L. Maher, Process models for design synthesis, *AI Mag.* 11 (4) (1990) 49–58.
- [24] B. Chandrasekaran, Design problem solving: a task analysis, *AI Mag.* 11 (4) (1990) 59–71.