Title

Cognition in Design Process

Permalink

https://escholarship.org/uc/item/42c2v2pb

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 11(0)

Author

Chan, Chiu-Shui

Publication Date

1989

Peer reviewed

Cognition in Design Process

Chiu-Shui Chan

Department of Architecture

Carnegie Mellon University

Abstract

The purpose of this research is to study the cognitive process in architectural design problem solving. It also will explore a cognitive structure (model) capable of representing the problem solver's cognitive behavior. The goal plan, schemata, perceptual-test, and generate-and-test are regarded as cognitive mechanisms that evolved in the problem solving process. They were observed in an experiment in which an experienced architectural designer was asked to do a residential design. Results from protocol analysis showed that an invariant cognitive structure could be built upon these cognitive mechanisms to explain the problem solving behavior. This cognitive structure (model) also provides a framework for future simulation.

Introduction

Architectural design problem solving was studied first by psychologists to understand the nature of ill-defined problems (Reitman, 1964, Simon, 1973). Later on, architectural researchers followed the same notion, and studied the cognitive aspects in solving architectural design problems. Among these studies, some used retrospective and introspective methods (Krauss & Myer, 1970, Darke, 1979) while others used protocol analysis (Eastman, 1969, Eastman, 1970, Akin, 1978, Akin, 1986). Each study yielded its own findings. For instance, Eastman explored the operators that caused moves between states in the design process (Eastman, 1970), and Akin used schemata to explain the general design procedure (Akin, 1986). Regardless of the various approaches, they all shared a common observation that cognitive process in design is a cyclic process of generating and testing solutions.

The pioneer studies reviewed provided some understandings about the design process which had not been explored before. However, the cognitive process in design is not fully understood, and it needs more exploration. Simon and Eastman, in particular, declared that ill-defined problems can be broken down into well-defined subproblems (Simon, 1973), and certain processes in ill-defined problems are similar to those used in well-defined ones (Eastman, 1969). Thus, a question arises as to whether those cognitive factors (i.e., knowledge representation, control structure, and generate-and-test) which exist in a well-defined domain, would still remain unchanged in an ill-defined domain, or if there are certain relationships that hold these factors together. It is difficult to find a report that addresses these questions. This research will address those questions and also explore a cognitive model that is capable of simulating the general problem solving behavior.

Fundamentals of Design Problem Solving

The basic concept for exploring design problem solving is similar to that used for well-defined problems. The essential approach is to view the design as if the process occurs in a problem space which consists of immense knowledge states. Since architectural design is unique, its problem space is considered as having three major components. These components are representations existing in the problem states. The first one is a set of design unit hierarchy. The design unit hierarchy is a tree-like structure that arranges design units from larger and more abstract units to more detailed ones.

The set of design units are physical elements of building components either given by the problem task or generated by the designer at any intermediate problem state. For example, a branch of the tree in the hierarchy of a residential design may consist of first floor unit, living room, fireplace, down to the hearth of the fireplace. Each unit is a component of the former one. This hierarchical tree structure explains why design units appear from abstract to detail level as the design progresses, in a top-down fashion.

The second component is a set of design constraints. Design constraints are certain requirements that must be fulfilled in order to design a design unit or a group of design units. The set of design constraints is also given by the problem task or is generated by the designer. For example, these might include information about the owner, site condition, climate condition, specification of design units, and some special design requirements. They are imposed by the problem and thus define the problem space.

The third component is a set of goals in which a designer finds an object that satisfies a set of constraints. These three components plus a set of operators, which are defined as anything that changes the problem states, determine the design problem space. Any change in these components will alter the problem space. From the designer's perspective, it can be explained that at any state in the space, the designer works on a design unit, and applies some knowledge to generate a solution that satisfies certain constraints. And all these actions are under the guidance of a particular goal.

Cognitive Mechanisms

The goals, design constraints, and the design unit hierarchy are retrieved from memory, and are results from the operation of certain factors. These factors, the subject matter of this research, trigger and strategically guide memory retrieval for solving the problem at hand. Since they are the driving forces that move states and produce or even alter cognitive behavior, they are termed cognitive mechanisms in this research. The following descriptions provide a general idea about each of these mechanisms, and they were empirically observed in an experiment that followed.

Schemata

In the domain of design problems, design knowledge is represented by a hierarchical semantic network (Akin, 1978, Akin, 1986). Since a designer must handle design units during the process of design, design units are subjects of the processing of design information. It is appropriate to represent nodes in the semantic network by design units, and design units are grouped by having related architectural functional relationships. A designer must have knowledge of the general components (design units) of a building as well as generic knowledge of what they are and how to design them. Therefore, it is assumed that a set of schemata which contains a large amount of design information is associated with design units in the semantic net. In the network, there is a set of schemata called design constraint schemata which provide declarative knowledge and procedural knowledge about the constraints. For example, climate is a design constraint. The designer must know that the winter breeze from northwest would bring cold into the building (declarative knowledge), and s/he should also know how to use the building mass to block the wind or to reduce the glazing size on the windward surface to minimize the heat lost (procedural knowledge). These pieces of knowledge are stored in the schemata, and a design solution is generated by the application of this knowledge.

Generate and Test

Design problems have the nature of generate-and-test cycles (Eastman, 1969, Akin, 1978, Darke, 1979). Each cycle has two mechanisms: a generator and a tester (Simon, 1973, Akin, 1986). The generator takes some input to generate a solution or solutions. The input is assumed to have three sources: (1) an evocation of schemata from memory that activates information stored in the short-term memory and then applies it; (2) a series of schemata instantiations, in which the generator applies the embedded rules in schemata to generate a solution; (3) a retrieval from memory of a pre-solution model, which the generator either accepts or modifies to generate solutions. After the generator generates a solution, the tester tests against a set of constraints.

Goal Plan

The goal plan is a hierarchical process that controls the sequence of operations. In design problems, designers have a general design method stored in their long-term memory called a general goal plan, which consists of a sequence of general goals to be accomplished. This goal plan is believed to be the key mechanism that converts an ill-defined problem into manageable size.

Perceptual-test

Since design solutions are accumulated from state to state, information presented in external display changes accordingly. A designer must gather information about the problem situation from time to time, and this is done by perception. Researches on perception in problem solving have dealt with the perception of chess positions (DeGroot, 1966, Simon, 1969), or solving the Tower of Hanoi puzzle (Simon, 1975). The perception has been formulated by production systems to describe the function of its mechanisms, and is referred to as perceptual-test (Simon, 1975). The perceptual-test will perceive the problem context and the solution context to determine the appropriate action to be executed next. Hence, it is regarded as the control mechanism in the design process.

Observations

In order to empirically test the existence and the function of these cognitive mechanisms, an experiment was conducted. The subject was a PhD student in architecture. He had eight years of architectural design experience at the time this experiment was conducted. The task was to design a three bedroom dwelling for a single family. Design units included a workshop, living room, dining room, bathroom and two bedrooms for a son and a daughter. The total floor area was limited to 2,200 square feet. The client was a professional architectural perspective draftsman. In order to discern the kind of design knowledge that the subject would retrieve from memory, the design information provided in the instruction was reduced to a minimum. Protocol data were collected for analyses. The methods for coding protocol, classifying episodes, verifying data, and developing problem behavior graph were described in detail in another report (Chan, in press). Results discovered in relating to the cognitive mechanisms were the following.

Constraint Schemata in Design

Constraints in design are of two sorts: global and local ones. Global constraints are applicable to a group of design units. Local constraints are bound to two or fewer design units. In this experiment, the subject retrieved a few global constraints, i.e. light, privacy, accessibility to the road, symmetrical disposition, room dimension, and land slope. These global constraints reflected the following characteristics: (1) they were mostly evoked during the first episode of the protocol data; (2) they were applicable to a group or to all design units; (3) they were able to be used in different tasks.

The subject retrieved 47 local constraints in this protocol, and they appeared only at the two lowest levels in the design unit hierarchy constructed from the data. The sequence of retrieving constraints reflected the following factors. At the early design stage, global constraints are evoked for the purpose of organizing the problem structure. Then, based on the structure developed, a design scenario is formed to guide the later design. As the design progresses, design units are handled from larger units to detailed ones and the associated constraints are retrieved accordingly.

Constraints are retrieved so that embedded knowledge can be applied to generating or testing solutions. An excerpt from the protocol of the subject demonstrates this:

> "Now, somehow, it seems that this (northeast) corner here, seems more private. Because these two edges (west and south) are bound by outside roads. And there is a property on this (north) side and a private property on this (east) side. So, things will be better, if I place things along this (northeastern corner) side."

These protocol statements can be converted into schemata representation as follows.

```
Schema A: <Site-privacy> (<Building>)
        Rule = If there is a <Private-corner>
                Then put building at <Private-corner>.
Schema B: <Private-corner>
        Rule = If <Private-edge> (<A>) is private
                and <Private-edge> (<B>) is private
                and <A> and <B> are adjacent
                Then the corner formed by A and B is a private corner.
Schema C: <Private-edge> (<X>)
        Rule = If <X> = next-to-a-property
                Then it is private.
```

A constraint schema as shown in the examples consists of an identifier, a variable, a set of rules, and a value of the variable. For instance, in schema A, the identifier is site-privacy, the variable is the <Building>. The factual knowledge of the private corner in the site is embedded in the left hand side of the production. The procedural knowledge, which is to put the building at the private corner, is at the right hand side. The value of the schema is obtained from evaluating the rules in the schema, as a result, the value is returned to the variable. As in this example, in order to satisfy the privacy constraint, a series of schemata, from A to C, were instantiated to generate the solution. This shows the concept of applying the schemata for solution generation.

Generator and Tester

The generator uses three sources of input. One is to instantiate a series of schemata and to apply a series of rules to generate solutions as described before. The second one is to retrieve pre-solution models from memory and apply these models. A pre-solution model is a design solution generated from experience. Since architectural design deals with graphic representation, design solutions are mainly images, and thus possibly stored in memory by some kind of image code (Chan, 1989). In this experiment, the subject used nine pre-solution models, and seven out of nine were iconic images. For example, in dealing with the porch roof, the subject could quickly retrieve a pitch roof image and draw it. This showed a recognition search method used by the generator. The recognition search involves reducing the problem to a point at which a known procedure or model can be applied to the remaining stages.

Besides recognition search, the generator also used the means-ends analysis method to accommodate a pre-solution model. This occurred when the subject retrieved geometric blocks with central cores (a pre-solution model) to solve the service core problem (the problem that contained the stair case, utility core, and bathroom facilities). But this model did not fit the context because it left no room for a corridor. The subject's strategy was to retrieve seven rules to gradually modify the pre-solution model until he achieved a satisfactory solution. His solution was to put the staircase next to the bathroom and locate them in two rectangular blocks with a corridor running in front. In this example of the means-ends analysis search method, a modified image was the end.

The third input source for the generator is to simply retrieve a constraint schema and apply its rule. This source also apply to the tester, and is regarded as the main characteristic of the generator and tester. The problem behavior graph constructed from protocol data showed that whenever a design solution was generated or tested, at least one design constraint was involved and at least one rule was used. This suggests that the knowledge in constraint schemata is the source for problem solving, and it also explains why design constraints are important in design tasks.

Goal Plan

The protocol of the subject's goal development in this experiment showed some clear distinctions between episodes. In transitions between goals, the new knowledge state did not correlate to the previous one. A new goal was developed and verbalized all of a sudden. This supported an argument that new goals are retrieved from the goal plan in memory. The subject's goal plan looked like this: task understanding, site organization, scenario development, initial space layout, room size arrangement, space generation, first floor layout, second floor layout, elevation, site development, and finally evaluation.

Perceptual-test

The perceptual-test controls the design process and has been observed to have four functions. The first function is to determine whether the current goal has been accomplished. In the protocol, the subject made no statement indicating that he had satisfied a particular goal. The subject simply proceeded from one goal to another. The silent switch suggests that unless a goal is achieved, it is impossible to develop a new one.

The second function is to test the generated solution to perceive the solution path. For example, one of the task requirements was to include a Doric column in the residential design. One of the generated solutions was to locate the Doric column in the center of a room as a single interior element supporting the ceiling. The subject indicated two things: (1) such a form must also match a classical vault, but this usage would change the character of the design; (2) the subject was not keen on doing a historical revival. Therefore, this solution was abandoned. This indicated that the subject perceived a critical problem situation at the time he generated a solution. The critical problem situation refers to the possibility of changing the problem structure or solution path.

The third function is to perceive what is lacking at the present stage and search for a design unit to work on next. Take an excerpt from the protocol for example: "I am trying to see in terms of section what is going to have, and I am trying to see what other things could be attached to this column. One thing is that, you may call it some kind of glazing, in which the column really is a free-standing element, visually." In this example, the subject searched for a new design unit to attach to the column in order to visually make the column a free-standing element. The new design unit he evoked was glazing on the side of the column.

The fourth function is to perceive the problem context and the solution context to determine the next action. Perception of the problem context means understanding the problem structure to determine the goal sequence. For example, the subject perceived the size of the building mass as a small one which would not affect its location on the site, so he decided to develop the site later. And it turned out that the goal of site development appeared at a later stage in the protocol. Perceiving the solution context means seeing the solution path and using it to generate the next solution. For instance, the subject used symmetry as a constraint, which developed while he was arranging the position of a bay window and a Doric column (the Doric column was on the central line of the bay window). This symmetry constraint was again selected later on to solve the living room layout. The subject indicated that "since it (Doric column) is going to be something as striking as an element like that (symmetric character), at least here (in the living room plan) I am trying to keep this (living room) spaces, and try to maintain the same symmetric disposition." This implies that the subject perceived the solution context and selected the next solution which had the best fit. The solution context means that the occurrence of solution B is related to solution A, or that the result of solution A leads to the cause of solution B.

Invariant Structure

Observations based on the protocol data supported hypotheses about the functions of the cognitive mechanisms. Since these mechanisms are major constituents of the system, they control the overall process of design and thus generate a skeleton of an invariant structure.

As shown in Figure 1, this invariant structure represents the cognitive process. It shows that a design task can be broken down by means of the sequence of goals. Goals are generated either from a goal plan that is stored in memory, or from a perceptual-test. The goal plan contains a sequence of goals that the designer must know in order to process the design task, and must achieve in order to get the design problem into the final goal state. In accomplishing a goal, the designer manipulates a set of design units. A package of knowledge about the design unit called a schema, which contains associated design constraints and rules for application, is stored in a knowledge base as a part of the designer's long-term memory. By taking a set of design units and retrieving its associated schemata, design solutions for a particular goal are generated and tested. By repeating the process (taking a goal, activating a design unit, retrieving a set of associated schemata, applying a rule to generate a solution and then testing the solution), the design problem gradually moves toward the final state.

Obviously, the perceptual-test occupies a control position in Figure 1. It is necessary to emphasize the role of the perceptual-test in the system. The perceptual-test serves the following functions:

1. The test of the goal state will guarantee that the system is always in progress and that the process always moves toward a goal. Thus, if the current goal has been achieved, then the

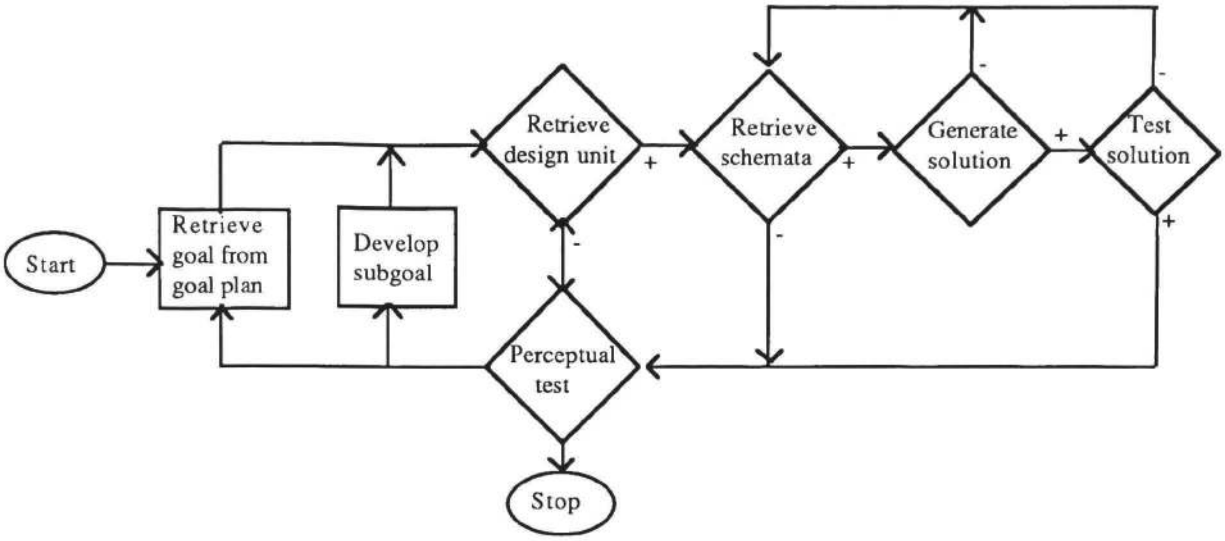

Figure 1: A cognitive model of design process

system will produce the next goal from the goal plan. Otherwise, the perceptual-test will perceive which design unit is the next candidate to continue accomplishing the current goal.

2. The test of global constraints will make sure that the generated solution is optimal. If the generated solution satisfies all the constraints, then the system will proceed to the next design unit under the current goal. Otherwise, a new goal is set up.

3. If a design unit is presented in short-term memory and a set of constraint schemata is evoked, the perceptual-test will recognize that such a design unit must be solved in order to process the next one. Thus, a subgoal is developed to solve the problem being presented.

4. The perceptual-test will perceive what happens at the current state and will determine the appropriate next step.

Conclusion

The cognitive mechanisms are regarded as fundamentals in processing information. Among them, the perceptual-test controls the process, and provides problem solving strategies. A set of production systems to account for the subject's strategy and control structure in this experiment had been explicitly developed and hand simulated elsewhere (Chan, in press). This set of production systems is inferred to be a prototypical template or program stored in the memory, and is instantiated at the time when a problem is encountered. The next step beyond this study is to implement the proposed computer simulation model of this process.

It is further inferred that the cognitive mechanisms studied are essential in solving ill-defined problems. The differences between individual human problem solvers are the *information* stored in the schemata, the *goals* in the goal plan, the *search methods* utilized by generator and tester, and the *control strategies* developed by perceptual-test. These factors can be termed as cognitive variables, which are the operational sources of the cognitive mechanisms and are important clues for studying the individual differences.

This study also provides a systematic approach for studying design processes that are recognized as a part of ill-defined problems. Although it is not certain that all kinds of ill-defined problem solving (music composition, painting, and story writing) have characteristics in common, this study suggests that ill-defined problems rely greatly on the problem solver's prior knowledge and control strategy for tackling problems.

The theory set up in this study is strongly supported by the data obtained from the experiment. Its further application is to study how style is generated from design processes, and to study what cognitive aspects would likely influence the formation of a style. Only after more experiments conducted on more subjects will the accuracy of the model be convincing.

Acknowledgement

The author would like to thank Herbert A. Simon, Omer Akin, and John R. Hayes for their discussions and comments on the theories being set forth in this research.

References

Akin, O. (1978). How do architects design? In J. C. Latombe (Ed.), *Artificial Intelligence and Pattern Recognition in Computer Aided Design*. New York: North-Holland.

Akin, O. (1986). *Psychology of Architectural Design*. London: Pion.

Chan, C. S. (in press), Cognitive Processes in Architectural Design Problem Solving. *Design Studies*.

Chan, C. S. (1989). Mental Image and Internal Representation. Manuscript submitted for publication.

Darke, J. (1979). The primary generator and the design process. *Design Studies*. 1(1), 36-44.

De Groot, A. D. (1969). Perception and memory versus thought: some old ideas and recent findings. In B. Kleinmuntz (Ed.), *Problem Solving*. New York: Wiley.

Eastman, C. M. (1969). Cognitive processes and ill-defined problems: a case study from design. *Proceedings First Joint International Conference on Artificial Intelligence*. Washington, D. C.: Joint International Conference on Artificial Intelligence.

Eastman, C. M. (1970). On the analysis of intuitive design processes. In G. T. Moore (Ed.), *Emerging Methods in Environmental Design and Planning*. Cambridge, MA: M.I.T.Press.

Krauss, R. L., & Myer, J. R. (1970). Design: a case history. In G. T. Moore (Ed.), *Emerging Methods in Environmental Design and Planning*. Cambridge, MA: M.I.T. Press.

Reitman, W. R. (1964). Heuristic decision procedures, open constraints, and the structure of ill-defined problems. In M. W. Shelley, & G. L. Bryan (Eds.), *Human Judgments and Optimality*. New York: Wiley.

Simon, H. A. (1973). The structure of ill-structured problems. *Artificial Intelligence*, 4, 181-201.

Simon, H. A. (1975). The functional equivalence of problem solving skills. *Cognitive Psychology*, 7, 268-288.

Simon, H. A., & Barenfeld, M. (1969). Information processing analysis of perceptual processes in problem solving. *Psychological Review*, 76, 473-483.