

## **APPENDIX 2: IS DESIGN “SEARCHING” OR “LEARNING”?**

The first paradigm through which researchers have studied and begun to understand design was introduced by Herbert Simon in the early 1970s (Simon 1973). In this paradigm, design is viewed as a rational search process: the design problem defines the “problem space” that has to be surveyed in search of a “satisficing” design solution. Seeing design as a rational problem-solving process entails adopting a positivistic view of science, taking natural sciences as the model for a science of design. The rational problem-solving approach to design is a combination of practice-based phase models of the design process and a model of the designer as an information processor from the field of cognitive psychology. The glue that holds these together is the theory of “human problem-solving.” The central paradigm in this field is that problem-solving can be described as “a search for a solution through a vast maze of possibilities [within the problem space]. ... Successful problem solving involves searching the maze selectively and reducing it to manageable proportions” (Simon 1969, 54).

These search processes have been studied through protocol analysis of subjects solving chess and cryptarithmic problems. They can be displayed and analyzed in “problem behavior graphs” (Newell and Simon 1972). Simon’s key contribution to design methodology was to state that the productive design thought could be captured in the same positivistic framework. Problem solvers are seen as “goal-seeking information processing systems,” operating in an objective and knowable reality. Simon explicitly states that his theory does not take into account the processes and results of human perception; it assumes that “human beings, viewed as behaving systems, are quite simple. The apparent complexity of our behavior over time is largely a reflection of the complexity of the environment in which we find ourselves.” In studying an “adaptive system” (like man), we can often predict behavior from knowledge of