Introduction

According to science philosopher Jacob Bronowski,[1] design is the epitome of intelligent behavior: it is the single most important ability that distinguishes humans from other animals. Although some animals can use tools to help them accomplish certain tasks—such as extracting termites from a mound, or breaking coconuts—no other animal is capable of analyzing a problem to uncover its root causes, which can help it to consistently and deliberately devise the means to solve the problem even when these means are not immediately obvious.

Problem analysis is a rational behavior—formal deductive, inductive, and abductive logical methods combine with experience-based heuristic reasoning to uncover the roots of a problem and indicate a course of action that will lead to its successful resolution. It relies on the problem solver's familiarity with formal reasoning methods and ability to frame the problem in a manner that will make it amenable to solution.

Problem analysis plays a major role in the process of design, but it is not the only ingredient. Unlike other problems—such as those posed by a game of chess—that rely solely on the problem solver's ability to reason and can, therefore, be solved through rational behavior alone, design problems are "ill-structured," according to Herbert Simon,[2] and downright "wicked," according to Horst Rittel.[3] They often do not contain enough information to be solved rationally, and they confront the designer with uncertainties that must, nonetheless, be dealt with. They typically must achieve multiple, often conflicting, goals, requiring the designer to make difficult tradeoffs whose outcome cannot be reliably predicted. And they always have side effects and aftereffects, which may render the solution unacceptable for reasons not directly associated with the problem itself.

To overcome these difficulties, designers must rely on intuition and creativity: the cognitive facilities of "knowing" without the use of rational processes, culminating in the celebrated "intuitive leap"—when all the pieces of the puzzle somehow seem to fall into place, and an overall order descends upon the problem. Neither of these facilities can be well defined, let alone codified or taught. They are innate abilities, which distinguish the artist from the mere artisan, the genius from the merely competent.

But unlike art, which must often conform only to the artist's self-imposed goals and constraints, architectural design is an activity that deals, in equal measures, with externally imposed constraints (e.g., site conditions, climate, functionality, cost, building codes, and so forth) and internally drawn inspirations. It thus relies on both sides of the brain—the analytical and the creative—to produce solutions to problems that cannot be solved with one facility alone.[4]

Computers, by their nature, are superb analytical engines. If correctly programmed, they can follow a line of reasoning to its logical conclusion. They will never tire, never make silly arithmetical mistakes, and will gladly search through and correlate facts buried in the endless heaps of information they can store. They will do all that quickly and repeatedly, by following a set of instructions called a program*,* which tells them in minute detail how to manipulate the electrical impulses in their circuits. They can present the results of these manipulations in the form most suitable for human comprehension: in textual reports, tables of numbers, charts, graphical constructions—even in dynamically changing images and sounds. But while they can follow instructions precisely and faultlessly, computers are totally incapable of making up new instructions: they lack any creative abilities or intuition.

What, then, is the use of computers for the process of design, which requires both rational and creative abilities, if they lack one of the two key ingredients needed to solve design problems? Why do we even bother to draft them into the service of designers? Is it because we humans, who possess both rational and creative abilities, are easily bored, distracted, and tend to make mistakes when confronted with large and complex problems? While our memories are vast enough to store the experiences of a lifetime, our ability to recall these memories at will is limited. This is precisely where computers excel. If we could find a way to take advantage of the abilities of computers where ours fall short, and use our own abilities where computers' fall short, we would create a very powerful symbiotic design system: computers will contribute their superb rational and search abilities, and we humans will contribute all the creativity and intuition needed to solve design problems.

Computers, for example, could list and keep track of all the goals and constraints the design solution must accomplish. They could group them into related issues,[5] search for precedents, even propose possible alternative standard solutions. The designer could then use these as the basis for developing new solutions that better fit the problem, which the computer could analyze and compare to the stored list of goals and constraints. Once a solution has been found, the computer could help represent it graphically and numerically and communicate it to other partners in the design process. It could then keep track of changes and updates, even alert the designer to potential inconsistencies and errors. Furthermore, computers could help fabricate and construct the resulting buildings, much like robotic machines now help fabricate cars, airplane parts, and integrated circuits. They could even help us manage the buildings once they have been constructed, much as they control the engine of a car or monitor elevators in buildings. And, further still, computers could provide an alternative "space" for human inhabitation—the so-called cyberspace—which could offer a new stage for human activities, from education to commerce to entertainment.[6]

Such a symbiosis is predicated on *communication*: the ability to share information between humans and computers. But communication, as discussed in part 2 of this book, is a process that relies on shared knowledge, which the communicating parties use to interpret the information. It is relatively easy to communicate information from computers to humans, who posses the intelligence needed to understand textual, numerical, graphical, and auditory messages. But it is frustratingly difficult to communicate information from humans to computers, who lack the intelligence and the ability to interpret messages, unless they are coded in a completely unambiguous manner. Communicating the nuances of an idea—especially a design idea—from humans to computers is, therefore, a very tall order. Although some attempts have been made to solve this problem,[7] most researchers have opted to avoid it by placing the entire design process within the computer's electronic realm.

Hence, the majority of computer-aided design research over the past fifty years has been directed toward developing computational systems that provide varying levels of assistance to human designers by taking care of smaller or larger parts of the design process. They range from drafting and modeling systems, where the role of computers is limited to supporting human designers in drawing lines and other geometrical entities that have no meaning to the computer; to analytical systems with enough "understanding" of the data to be able to provide rational appraisal of human designers' solutions (e.g., energy, cost, fire egress, acoustics, and so forth); to knowledge-based, "intelligent" design systems that can actually propose design solutions for appraisal and further development by human designers. Along the way, systems have been developed that offer design information storage and query capabilities and systems that help human designers communicate with one another. Each type of system has found its niche and provides useful service to its users, but because of their widely different objectives, these systems can rarely communicate with one another, although attempts have been made since the 1960s to develop interoperability protocols.

The following chapters discuss the nature of design, the nature of computers, and how the two have been combined in the form of computer-aided design systems over the past fifty years.