A Framework for Design Problem-Solving

B. Chandrasekaran
Laboratory for AI Research, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio, USA

Abstract. We develop a task-structure for design problem-solving. The task-structure of a complex problem-solving activity such as design is a hierarchical organization of subtasks. For each task in the task-structure, we can then proceed to investigate what methods may be available, and what knowledge and inference requirements each of these methods have. Some of the methods may be domain-specific, some of them more generic in character, some may involve traditional computational techniques, and some others may involve searching in a problem space for solutions to the task. However, this systematic process of identifying tasks, methods, and subtasks will help us to see how design as a general problem is solved not by one method or technique but by an opportunistic exploitation of whatever methods are available (i.e., the knowledge required for using a method is available) to help accomplish a subtask. Thus, in principle, very different methods and knowledge can be brought into play in as flexible a way as applicable.

This work was motivated by the need to make sense of numerous proposals for design problem-solving in AI. It was clear that many of the methods or techniques “worked,” (i.e., produced solutions to some instances and versions of the design problem), but how these methods related to the general problem of design, and why and how the method worked in relation to the type of design problem that was being solved, were often not clear. Often, characterizations were given of the methods that were inappropriate, by confusing a relatively unimportant implementational aspect of the method with the more essential strategy that was encoded using that implementation (e.g., “design by object-oriented programming”, “design by assumption revision”). (See [5] for our earlier critiques of the confusion between the task level and the implementation level in the description of problem-solving systems.)

1 What Is the Design Problem?

The goal of this paper is to develop a task-structure for design as an information processing activity. The task-structure describes the possible ways in which the overall task can be decomposed into subtasks, which in turn can be similarly analyzed until tasks which are “primitive” in some sense are reached. For each task in the task-structure, we can then proceed to investigate what methods may be available, and what knowledge and inference requirements each of these methods have. Some of the methods may be domain-specific, some of them more generic in character, some may involve traditional computational techniques, and some others may involve searching in a problem space for solutions to the task. However, this systematic process of identifying tasks, methods, and subtasks will help us to see how design as a general problem is solved not by one method or technique but by an opportunistic exploitation of whatever methods are available (i.e., the knowledge required for using a method is available) to help accomplish a subtask. Thus, in principle, very different methods and knowledge can be brought into play in as flexible a way as applicable.

Design as a Task and as a Process

Design as a process and design as a task need to be distinguished. Design is a task in which the agent is given a set of primitive objects or a generator of such primitive objects, and ways of, and constraints on, combining them, and the aim is to put together an assembly of primitive objects selected from this set that satisfies a specified set of desired proper-
ties. Because of this, design is often said to be a problem of synthesis. Design covers a wide variety of phenomena: planning a day’s errands, theory construction in science, and composing a fugue are all design tasks. In fact, all of problem-solving is design (viz, synthesizing a sequence of operators in a state space which will transform an initial state into a goal state). For example, diagnosis can be viewed as a design task in which a diagnostic hypothesis is being composed.

Design as a process is the subject matter of design research. The investigation centers on identification of problem-solving processes which are particularly appropriate for design. A major paradigm for understanding problem-solving is search. Design as search implies a space of possible design assemblies in which a search process works. Theories of design then would propose particularly interesting organizations of the search space and control of search. Essential elements in a process being recognized as a design process are: components or partial designs being composed or assembled into larger subsystems, and some criterion of evaluation being applied to such assemblies. In some sense the process should be seen as a search in the space of possible subassemblies of components.

In contrast to the search view is the view of design as an “intuitive,” almost instantaneous, process where a design solution comes to the mind of the designer. Artistic creations, scientific theories, etc., are often said by their creators to have occurred to them in this manner. The role and forms of search, if any, in such unconscious processes in humans are open issues in cognitive science. Theories in AI that have anything at all to say about them are theories of memory retrieval, such as in case-based reasoning, but as of now such theories are much simpler than needed to account for complex design and discovery processes. In any case, even when such design proposals arise from such unconscious, apparently nonsearch processes, the proposals are still evaluated, critiqued, and modified in a deliberative manner. These critical and modification processes can be cast in the search paradigm for design, since they contribute to exploration of alternative designs in a design problem space. This paper discusses memory processes as contributing to the overall design process (as in case-based reasoning), but describes the tasks and subtasks in design largely at the level of deliberative problem-solving.

A design task may be performed by processes which are not particularly recognizable as design processes. Examples are a design problem which is solved by table look-up (where the table contains design specifications and an appropriate design for each specification), and a design problem which has been converted into a parameter selection problem, solvable by evaluating algebraic formulas. Often systems solving multiple-hypothesis diagnostic problems have “design” components which perform a very specialized form of hypothesis assembly [14].

The point of the above discussion is that whether a given task is a design task or whether a problem-solving process is a design process is not formally well-defined. However, intuitively, design as a task or process involves composing components into subsystems or subassemblies that cohere from the point of view of desired functions and constraints. In order to make progress in investigating the problem-solving processes involved in design, let us restrict the scope in this discussion to the design of artifacts that satisfy some functional specifications.

2.1 Functions and Constraints

Functions of an artifact are what we want the artifact to do or be. They can be expressed as a state or a series of states that we want the device to achieve or avoid under specified conditions. Functions may be explicit or implicit [17]. An example of an implicit function in many engineering devices is safety (e.g., a subsystem’s role can only be explained as something that prevents leakage of a potentially hazardous substance), and this function may not have been stated explicitly as part of the design specification.

Usually, design specifications will mention, in addition to desired functionalities, a number of constraints. The distinction between functions and constraints is not a formal one (i.e., functions are constraints on the behavior or properties of the device). It is, however, useful to distinguish functions from other constraints, since the former are the primary reason why the device is desired. Design constraints can be on the properties of the artifact (e.g., “Should not weigh more than . . .”), on the process of making the artifact from its description (manufacturability constraints), on the design process itself (e.g., “I want a design within a week”), and so on. For example, transportation is a desired function of a bicycle, while “light-weight” is a constraint. The reason why this distinction is important is that starting with design proposals which more or