1 Framing the Issues

The principal thesis of this book is that the key element of design is *representation*. If we were to consult a standard dictionary, we would find *representation* defined as "the likeness, or image, or account of, or performance of, or production of an artifact." Note, however, that whereas our dictionary defines representation as a *noun* in which terms such as *image* and *likeness* refer to the artifact being designed, it also suggests aspects of a *verb* when it defines the design process in terms of a performance or a production. This suggests that representation in design incorporates both representation of the *artifact* being designed as well as representation of the *process* by which the design is completed. We now examine briefly both types of representation.

1.1 Representation of Artifacts for Design

Suppose we are charged with the design of a safe ladder. What does it mean, first of all, for a ladder to be "safe"? That it should not tip on level ground? That it should not tip on a mild slope? What is a mild slope? How much weight should a safe ladder support? Of what material should it be made? How should the steps be attached to the frame? Should the ladder be portable? What color should it be? How much should it cost? Is there a market for this ladder?

We have quickly identified several – but by no means all – of the questions in this very simple design problem, and we would not be able to answer most of them by just applying the mathematical models that originate in the engineering sciences. For example, we could use Newton's equilibrium law and elementary statics to analyze the stability of the ladder under given loads on a specified surface, and we could write beam equations to calculate the bending deflections and stresses of the steps under the given loads. But which equations do we use to define the meaning of "safe" in this context or to define the color or the marketability? In fact, which equations do we use to describe the basic function of the ladder? We know that the function the ladder serves is to allow someone to climb up some vertical distance, perhaps to paint a wall, perhaps to rescue a cat from a tree limb, but it is for the designer to translate these verbal statements of function into some appropriate mathematical models at some appropriate time – often early on – in the design process. That is, we have no mathematical models that describe function directly; we infer functional behavior by reasoning about results obtained from manipulating mathematical models.

Thus, we recognize already that a *multiplicity or diversity of representations* is needed for design, a collection of representation schemes that would enable description of those issues for which analytical physics-based models are appropriate; those that require geometric or visual analysis to reason about shape and fit; those that require economic or other quantitative analysis; and those requiring verbal statements not easily expressed in formulas or algorithms. Some of the verbal requirements could be statements about *function*; about *form* (e.g., a stepladder or an extension ladder); about *intent* (e.g., to be used at home or to be used in an industrial environment); or about legal requirements (e.g., to satisfy government regulations).♦

> Verbal statements are also made to describe or reference *behavior* (e.g., steps on the ladder should not have too much give), *heuristics* (e.g., my experience suggests that fiberglass ladders feel stiffer than aluminum ladders), *design decision alternatives* (e.g., Can I choose between a stepladder and an extension ladder?), *preferences* (e.g., I like ladders that are bright blue), *affordances* (e.g., steps spaced in small, uniform increments enable climbing), *constraints* (e.g., the ladder can cost no more than $100), *assumptions* (e.g., I thought it would fit in my trunk), and *intent* or *rationale* (e.g., the ladder is for home use).

In essence, representation is modeling. However, representation in design is much broader than modeling in engineering science, wherein mathematical modeling is the key idea. In fact, a more apt analogy may be found in the linguistic notion of abstraction ladders or in Korzybski's aphorism, "The map is not the territory." The real point, however, is that we must represent meaningfully much more knowledge than can be set into mathematical formulas or numerical realizations of those formulas, and this is now possible. Advances in computing generated by AI research allow – and even encourage – the representation of symbols and thus objects, attributes, relationships, concepts, and so on. New programming styles have emerged in which we can capture more abstract conceptual and reasoned design knowledge that cannot be reduced to conventional algorithms.

Although we have discussed the role of representation in design before, we are not the first nor will we be the last to stress the importance of representation. As we noted previously, representation is one of the seven subjects in Simon's ideal design curriculum. (The other six subjects in the curriculum, in which Simon describes design as the science of the artificial, are evaluation theory, algorithms and heuristics for identifying optimum and satisfactory designs, formal logic for design, heuristic search, resource allocation for search, and the theory of structure and design organization.) A brief sampling of recent design research in which representation figures prominently includes work on features in mechanical design; shape grammars; object-oriented data structures; interaction of form, function, and fabrication; formal theories of design; shape emergence; and so on. We see from this list that the line between representing artifacts and representing the design process is not a sharp one. We will have a chance to explore some of this and related work in Chapters 5 and 6.

It is also important that we recognize that representation is not an end in itself but rather a means to an end; it is a way of setting forth a situation or formulating a problem so that we can find efficiently an acceptable resolution to a design problem. This also implies that representation is strongly coupled to whatever strategy we have chosen for solving a problem (whether in design or in some other domain). Because it is pointless to invoke alternate representations unless we gain some leverage thereby, the notion of changing the representation of a problem is inexorably linked to the idea that there is a problem-solving strategy available to us that uses this representation in a beneficial way. This is not to say that the research objective of developing new representations should be limited to those for which a problem-solving strategy is available. Research on both artifact representation and problem solving should proceed independently, although perhaps in parallel. But the development of new representations does suggest that broader paradigms of problem solving should be integrated into the outlook of engineers and designers, the idea being that approaches such as AI-based paradigms and tools will become part of the arsenal of weapons available for better engineering.

1.2 Representation of the Design Process

Let us return to the ladder-design problem, now with a view toward examining the process by which the design will be done. First, we recognize that the initial statement of the client's wishes is rather vague, in large part because it is simply a brief verbal description. In fact, design projects often originate with a brief verbal statement, such as President John F. Kennedy's lunar challenge. To proceed with a design, we have to flesh out these skimpy skeletons by *clarifying* and *translating* the client's wishes into more concrete objectives toward which we can work. In the clarification step, we ask the client to be more precise about what is really wanted by asking her or him questions: For what purposes is the ladder to be used? Where? How much can the ladder itself weigh? What level of quality do you want in this ladder? How much are you willing to spend? However, the degree of precision that we might demand from the client could well depend on where we are in the design process.

Some of the questions we asked in the hope of clarifying the client's wishes obviously connect with our previous discussion (cf. Section 1.1) on artifact representation, but some lead us into a process in which we begin to make choices, analyze the dependencies and interrelationships between possibly competing choices, assess the trade-offs in these choices, and evaluate the effect of these choices on our overall goal of designing a safe ladder. (There are formal methodologies for identifying trade-off strategies.) For example, the form or configuration of the ladder is strongly related to its function: We are more likely to use an extension ladder to rescue a cat from a tree and a stepladder to paint the walls of a room. Similarly, the weight of the ladder will certainly have an impact on the efficiency with which it can be used to achieve its various purposes: aluminum extension ladders have replaced wooden ones largely because of the difference in weight. The material of which the ladder is made not only influences its weight; it also is very influential in determining its cost and even its feel. Wooden extension ladders are considerably stiffer than their aluminum counterparts, so users of the aluminum versions have to get used to feeling a certain amount of "give" and flex in the ladder, especially when it is extended significantly. Thus, a possible design goal that was not even mentioned has suddenly emerged: design a safe, *stiff* ladder.

In the translation step, we convert the client's wishes into a set of *design specifications* that serve as benchmarks against which to measure the performance of the artifact being designed. The translation process is where the "rubber begins to meet the road," for it is here that the verbal statement is recast in terms of more specific design objectives. These specific objectives can be stated in a number of ways, reflecting variously the desire to articulate specific dimensions or other attributes of the designed object, which are usually called *prescriptive specifications*; specific procedures for calculating attributes or behavior, which are embedded in *procedural specifications*; or the desired behavior of the device, which is encoded in *performance specifications*. A successful design is one in which performance meets (or exceeds) the given specifications and satisfies (or exceeds) the client's expectations. We reiterate that the specifications may evolve or be further detailed and refined as the design unfolds. The role of specifications in design has been the subject of much thought and discussion, and it is also clear that the techniques for stating design specifications (and, later, fabrication specifications) are intimately related to design-representation issues.

The design process is evolutionary in nature, and we will come across choices to make and different paths to follow as a design unfolds. In fact, the particular choices that present themselves often become evident only after we have refined the original design objective – the client's statement – to some extent. For example, at some point in our ladder design, we have to confront the issue of fastening the steps to the ladder frame. The choices will be influenced by the desired behavior (e.g., although the ladder itself may flex somewhat, it would be highly undesirable for the individual steps to have much give with respect to the ladder frame) as well as by manufacturing or assembly considerations (e.g., would it be better to nail in the steps of a wooden ladder, or use dowels and glue, or perhaps nuts and bolts?).♦ Thus, the choices themselves need to be articulated in some language naturally conducive to making them; that is, choosing a particular bolt and nut pair to achieve a certain fastening strength requires access to a manufacturer's catalog as well as to the results of calculations about bearing and shear stresses. The particular process involved here could be called *component selection*, and it is invoked after we have *decomposed* the form of the ladder into its components or pieces, and after we have selected a particular type of component.

> Sometimes (e.g., in Gero's (1990) simple and often-cited model of design), the design process is driven by comparison between the design's expected behavior, derived from the desired function, and the predicted or actual behavior, resulting from the design's structure.

We are using this very simple example to illustrate the *formalization* of the design process through which we make explicit the ways we are doing some elements of our design. We could say we are *externalizing* aspects of the process so that we can move them from our heads into some recognizable language(s) for further analysis. There is no shortage of attempts to externalize design engineering processes, and we review many of these process models in Chapter 3. These descriptions and prescriptions are externalized to the extent that we can draw flow charts to describe the major steps of a design process, and the descriptions do point to analyses that need to be done and choices that must be evaluated, some of which can be done with conventional algorithms. However, these descriptions cannot be made computable because they are all relatively abstract; that is, they are not refined enough or rendered in sufficient detail that we can identify the underlying thought processes. Again, the objective in refining these processes is not just to be able to render them computable; it is to be able to *analyze* them in sufficient detail that we can *synthesize* design processes out of their fundamental constituent processes. When we do so in earnest in Chapter 6, we will see that we are taking advantage of research in AI (and related fields such as cognitive science) to examine and describe the activity that is called *design*. We view this as the *representation of the design process* as opposed to the representation of the artifacts that are being designed.

A recent knowledge-based system that illustrates the capture of a design process is called PRIDE; it serves as a designer's assistant for the mechanical design of paper-handling subsystems in copiers (see Chapter 6). Designing paper-transport systems for copiers is difficult because of the number and kind of design variables and their complex interactions. Nonetheless, by identifying the way designers actually do this task, the designers of the PRIDE system built a knowledge-based system that does much of the same design task as human designers. That is, PRIDE uses a variety of representation formalisms to incorporate both algorithmic and heuristic aspects of the design problem. It also uses a variety of *inference schemes* (i.e., reasoning patterns) and a powerful graphics interface to achieve a relatively complete simulation of the way human designers actually design paper-handling subsystems for copiers. The PRIDE environment allows the designer to experiment with different designs, both graphically and procedurally, and it facilitates the tracking of dependencies between design decisions and the maintenance of multiple design paths. The PRIDE system replicates a designer's approach to a complex problem in a way that simply cannot be done in a conventional, numerically based algorithm.

Furthermore, the PRIDE system works so well that it allows experienced designers to do feasibility studies in just a few hours, whereas it used to take four weeks to develop similar designs. In addition, the PRIDE system is viewed as useful because it also has led to paper-copier designs that are both more consistent and of higher quality.

We note two related points. First, the kind of replication or modeling of a design process that is found in the PRIDE system cannot be achieved by simply extending the traditional engineering science approaches to incorporate the thinking and logic characteristic of operations research (OR), as has been implied by some. The reason for this is that the representations inherent in OR approaches, although they permit the inclusion of economic or similar performance metrics, do not admit those qualitative or strategic choices that cannot be reduced to numbers.♦

A similar critique can be made about the use of *decision theory* (referred to as *decision-based design* (DBD)) because it requires knowledge of probabilities and utilities (see the Decision-Based Design Open Workshop Web pages (DBD 2004)).

The second point is that researchers in other engineering domains (recall that PRIDE's domain is mechanical engineering) have also clearly recognized the utility that knowledge-based (expert) systems have for modeling many phases of the design process – for example, in chemical engineering.

The second line of argument supportive of what has been outlined is that whereas much of the work in design is empirical in nature, both in design practice and in design research, there is apparently no objective basis for describing and evaluating experiments in design. Much of what is known and transmitted about *how* to design artifacts is – or is perceived to be – anecdotal in nature. To the extent that design knowledge is viewed as design *lore*, both the development of the discipline of design and its acceptance by the engineering community as a serious discipline with a rigor and logic of its own are inhibited. Thus, in this context as well, it could prove useful to adopt the relevant terminology and paradigms from AI and related cognitive fields, subjects that are themselves highlighted by experimentation and empirical development. One example is the technique of protocol analysis, which may be described as the process of organizing, understanding, and modeling verbal reports and analyses. This technique has been applied formally and informally to elicit and organize the knowledge that designers use in their own domains. The use of a formal structure and methodology in this particular context is bound to be beneficial in developing a communicable understanding of the process of design.

1.3 An Illustration from Structural Engineering

To illustrate the importance of representation in design and the diversity of representations that we actually use for artifact and process representation, we present now a brief discussion of the structural engineering problem. In essence, the problem is as follows (Figure 1.1). A structural need is identified, whether it is for a mill building or a concert hall. Then we choose a structural concept, perhaps a simple steel frame and steel roof truss for the mill building, something considerably more complex for the concert hall, and we move to preliminary design. In this stage, we usually restrict our efforts to rough sizing of the principal structural members, the object being to see whether the type of structural system that we have chosen is practically feasible. We then move on to flesh out the structure by estimating the types and sizes of the remaining members (e.g., in the mill building, purlins for the roof truss and floor joists as needed). Then we home in on the final, detailed design in which we calculate actual dimensions and placements for all members and their connections. In the final step, we check to ensure that our design meets all statutory requirements, including both applicable building codes and design codes

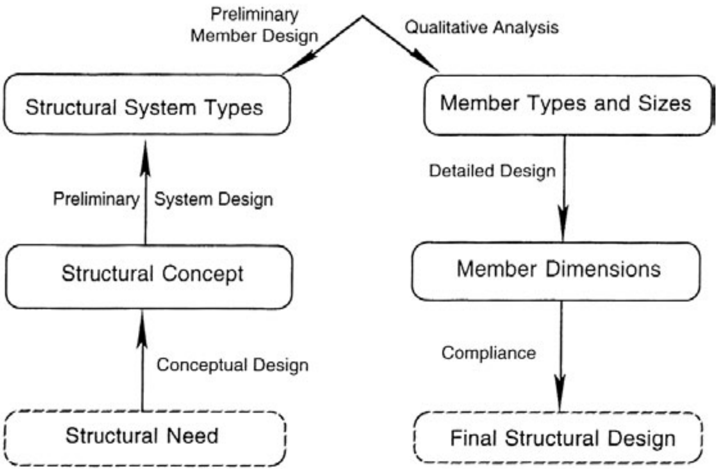

Figure 1.1. A pictorial view of the structural engineering problem (Fenves, 1993).

such as that of the American Institute of Steel Construction (AISC), which lays out performance specifications for steel members and connections.

Let us now examine the kinds of design knowledge deployed in completing such a structural design. Among the kinds of knowledge we apply are classical mechanics (e.g., Newton's laws); structural mechanics (e.g., models of columns and beams); geometry of structures (e.g., relating the geometry of members and assemblages of members to the orientation of the loads they are expected to carry); structural-analysis techniques (e.g., moment distribution for frames and the method of sections for truss analysis); behavioral models (e.g., modeling the stiffness of a complete frame); algorithmic models of structures (e.g., finite element method (FEM) computer codes); structural design codes (e.g., the AISC code); heuristic and experiential knowledge, both derived from practice and encoded in specifications; and *meta-knowledge* about how and where to invoke the other kinds of knowledge. Much of this knowledge is multilayered. For example, our understanding of the behavior of structural systems is realized at three distinct levels: spatial layout (e.g., where to place columns to achieve clear floor spans), functional (e.g., how to support different kinds of loads), and behavioral (e.g., estimating the lateral stiffness of frames).

How do we represent these different kinds of structural design knowledge? In fact, we use several different kinds of representations of the knowledge itself, including mathematical models for classical and structural mechanics (e.g., partial differential equations and variational principles); case-specific analyses (e.g., buckling of slender columns); phenomenological, "back-of-the-envelope" formulas (e.g., the beam-like response of tall buildings); numerical programs (e.g., FEM codes); graphics and computer-aided design and drafting (CADD) packages; rules in design codes (e.g., the AISC code); and heuristic knowledge about structural behavior, analysis techniques, and so forth. Such qualitative knowledge is often subjective and frequently expressed in rules. Thus, we already employ several different representations or "languages" of knowledge, including verbal statements, sketches and pictures, mathematical models, numerically based algorithms, and the heuristics and rules of design codes. When we use these different languages now, we manage to choose (in our head) the right one at the right time; however, in computational terms, we should recognize that it would be desirable to link these different representations or "languages" so that we could model our design process in a seamless fashion. We should also recognize that we often cast the same knowledge in different languages, depending on the immediate problem at hand. For example, a statement (typical of that found in building codes) that the deflection of a floor in a residential building should not exceed its length (in feet) divided by 360 is actually a restatement of equilibrium for a bent beam.

The point we want to make with this example is that for "real" engineering design problems – although it is equally true of our "toy" problem of ladder design – we are already accustomed to handling very complex representation issues. What is beginning to be true now is that we want to formally recognize this in the increasingly elaborate computer-based design tools we are developing. And, even more important for our present purpose, as we try to externalize our design knowledge, we are increasingly conscious of how we think about design. It is this raised consciousness we seek to expose here.

1.4 On the Role of Computation

The final argument we make in this book is, comparatively speaking, relatively straightforward. The rapid advances in the field of computer science, in both software and hardware, have brought increasing opportunities – and pressures – to "computerize" and automate engineering practice as much as possible and, at the very least, to automate the tedious and repetitive parts of engineering.♦ We take it as obvious that there are different opportunities for automation in different domains and for different tasks and task types within domains. For example, it has been easier to develop knowledge-based (expert) systems to perform *derivation* tasks, in which assessments are derived from data, than *formation* tasks, in which we attempt to form results to meet specified goals. Similarly, in exploring applications of AI techniques to design, there are going to be differences that ought to be acknowledged from the outset. For example, truly routine design (which is essentially a repetitive process) is much more readily automated than nonroutine design, in which the form and function (or their attributes) of a successful design may not be easily described, if at all. Thus, replication of routine design will offer different opportunities for automating with AI techniques than will the modeling of creative or original design. That is, it is likely that over time, a hierarchy of design tools will be developed to reflect these differing design tasks.

> Current computational resources provide design support by also allowing complex simulations of forces and flows, visualizations of those data, animations of structural models, virtual manufacturing, and rapid prototyping.

However, the perception of what may be automated – as opposed to what may be encapsulated in a designer's assistant – should not be perceived in static terms. As we articulate and acquire design knowledge, which we must do before we can represent it, we also acquire a keener understanding of that knowledge.♦ This results in a new consciousness of that knowledge, which in turn lays the foundation for discovering new algorithms, new procedures and strategies, or even new representations that may allow more of the process to be automated. Furthermore, as we noted earlier, the boundary between what we can understand and model as a cognitive process and true creativity is a shifting one, and we should not at this point preclude any endeavor that might prevent us from moving that boundary closer to the edge of complete understanding.♦ Still, the goal is not automation of the entire design process; it is the automation of the routine and the boring, and the creation of computer-based tools that facilitate design exploration.

A final note on computation. We have argued that the mathematics that we use to describe and analyze many engineering problems is inadequate for describing and analyzing many attributes of designed artifacts and design processes. Thus, we need to augment our mathematical modeling tools with others, such as graphics, logic, grammars, word and document processors, and – most relevant to this discussion – those tools based on symbolic representation. We must caution, however, that we are *not* saying that there is no mathematical foundation underlying the symbolic-representation techniques the use of which we advocate. Indeed, there are very complex mathematical problems involved in computation in general and in developing the underlying structure of the kinds of AI programs that are used to develop the kinds of results that we will see later in this book. However, the mathematics involved there is concerned with representing the symbols and the processes used to compute with these symbols so that, ultimately, the computer can do as it is told. Perhaps a very loose analogy is that this particular kind of mathematics is to the symbolic representation that we espouse as set theory and functional analysis are to the continuous mathematical models we routinely employ (e.g., the partial differential equation governing the bending deflection of a plate). Thus, we view as parallel the descriptive representations offered by continuous mathematical models and by symbolic representation of physical and conceptual objects and their attributes and dependencies.

> The R1/XCON (McDermott 1982) and PRIDE (Mittal and Dym 1985; Mittal et al. 1986) projects were among the first to note that the knowledge-acquisition process caused knowledge to be articulated that had not been previously recorded (i.e., the acquisition process was worthwhile even without the resulting configuration system). In addition, it is worth noting that an attempt to program the R1/XCON configuration design process as an algorithm failed, whereas using an AI-based technique (rules) for representing knowledge was successful.

> Recent research has focused on computational design creativity, arguing that creativity is not a mystery and that it can be studied scientifically and investigated computationally (Boden 1994; Brown 2008).

1.5 Bibliographic Notes

The dictionary definition of *representation* is from Woolf (1977).

Section 1.1: Mathematical modeling is discussed in Dym (1983) and Dym and Ivey (1980).♦ Korzybski is quoted in Hayakawa (1978). The new, AI-based programming styles are thoroughly described in Bobrow and Stefik (1986) and Stefik and Bobrow (1985). We have discussed representation before in Dym (1984) and Dym (1992b).♦ Simon's ideal design curriculum is outlined in the classic Simon (1981).♦ Recent design research topics include features in mechanical design (Dixon 1988; Dixon, Cunningham, and Simmons 1989); shape grammars (Stiny 1980); object-oriented data structures (Agogino 1988a); interaction of form, function, and fabrication (Rinderle 1985; Rinderle et al. 1988); formal theories of design (Fitzhorn 1988; Stiny 1988a); and shape emergence (Gero and Yan 1993). A good snapshot of modern AI-based research is Tong and Sriram (1992a, 1992b).

> Mathematical modeling also is discussed by Dym (2004).

> For a good text about knowledge representation and reasoning, see Brachman and Levesque (2004).

> Recent textbooks that suggest material for inclusion in a design curriculum include Dym et al. (2009), Ullman (2009), and Ulrich and Eppinger (2007). Design education continues to be a subject of discussion and research: for example, see Dym et al. (2005), the Web pages of the biennial series of Mudd Design Workshops (MDWs), as well as special issues of the *Journal of Mechanical Design*, "Design Engineering Education" (Doepker and Dym 2007), and *AIEDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, "Design Pedagogy: Representation and Processes" (Frey et al. 2010).

Section 1.2: Wood and Antonsson (1989, 1990) discuss the role of precision in design. Otto and Antonsson (1991) present a formal methodology for identifying trade-off strategies. The role of specifications in design is discussed in Fenves (1979); Stahl et al. (1983); and Wright, Fenves, and Harris (1980). Techniques for stating design specifications and their relationship to design-representation issues are described in Dym et al. (1988); Garrett and Fenves (1987, 1989); and Garrett and Hakim (1992). "Externalized" models of the design process are given, for example, in Dixon (1966), Woodson (1966), Jones (1970), Pahl and Bietz (1984), Cross (1989), French (1992), and Ullman (1992a, 1992b). Discussions of the PRIDE system include the original paper (Mittal, Dym, and Morjaria 1986) and a retrospective view (Morjaria 1989); see also Section 6.3 and its citations. An interesting view of the role of operations research in design is that of Wilde (1988). Applications of knowledge-based systems to chemical engineering are offered in Lien, Suzuki, and Westerberg (1987). Protocol analysis is defined in Ericsson and Simon (1984); its application to design has been explored formally in Stauffer, Ullman, and Dietterich (1987) and Ullman, Dietterich, and Stauffer (1988) and informally in Mittal and Dym (1985).

Section 1.3: A steel design code can be found in AISC (1986). The types of knowledge deployed in structural engineering are discussed in Dym and Levitt (1991b).