UNDERSTANDING THE CLIENT'S PROBLEM

What does this client want?

IN THE preceding chapters we have defined engineering design, explored and described the process of design, and detailed some tools that we can use to monitor and control a design project. Now we turn to describing the tools we use in the preprocessing phase of design, during which we are working toward developing an engineering definition of the problem. The set of activities done during the preprocessing phase is also known as *problem definition*.

3.1 OBJECTIVES TREES: TRANSLATING AND CLARIFYING THE CLIENT'S WANTS

The starting point of most design projects is the identification by a *client* of a *need* to be met. The fulfillment of that need then becomes the goal of the chosen design team. As depicted in the models of the design process in Figures 2.1–4, the client's need is quite often presented as a verbal statement in which the client identifies a gadget that will appeal to certain markets (e.g., a container for a new beverage), a widget that will perform some specific functions (e.g., a chicken coop), or a problem to be fixed through a new design (e.g., a new transportation network and hub).

Sometimes clients' project statements are quite brief. For example, consider the beverage container problem described in Section 2.5. Whether the design team is working for Great American Food and Tobacco (GRAFT) or for Bringing Juice Into Children (BJIC), it might simply get a memo from upper management that says: "Design a bottle for our new children's fruit juice product." The design team could easily respond to this directive by choosing an existing bottle, slapping on a clever label, and calling its work done. However, we might ask whether this new bottle is a *good* design, or, further, whether it's the *right* design. Answers to these questions depend on how we measure goodness and on how we assess rightness or correctness. Thus, we will also discuss metrics against which we can measure designs in this chapter.

Another simple project statement might take the form of "The Claremont Colleges need to reconfigure the intersection of Foothill Avenue and Dartmouth Avenue so students can cross the road." While communicating someone's idea of what the problem is, statements like this one have limitations because they often contain errors, show biases, or imply solutions. *Errors* may include incorrect information, faulty or incomplete data, or simple mistakes regarding the nature of the problem. Thus, the problem statement just given should refer to Foothill Boulevard, not Foothill Avenue. *Biases* are presumptions about the situation that may also prove inaccurate because the client or the users may not fully grasp the entire situation. In the traffic example, for instance, the real problem may not be related to the design of the intersection but to the timing of the signal lights or to the tendency of students to jaywalk. *Implied solutions*, that is, a client's best guesses at solutions, frequently appear in problem statements. While implied solutions offer some useful insight into what the client is thinking, they may wind up restricting the design space in which the engineer searches for a solution. Also, sometimes the implied solution fails to actually solve the problem at hand. For example, it is not obvious that reconfiguring the intersection will solve the student traffic problem. If students jaywalk, reconfiguring the intersection will do little or nothing to mitigate this. If the problem is that students are crossing a dangerous street, we may want to relocate the destination to which they are headed. The point is that we must carefully examine project statements in order to identify and deal with errors, biases, and implied solutions. Only then do we get to the real problem.

The client's understanding of the problem usually requires clarification by the designer.

For now, we want to focus on developing a clearer understanding of what the client wants because this will help us see the lines along which measures for a design might emerge. That is, we want to clarify what the client wants, account for what potential users need, and understand the technological, marketing, and other contexts within which our gadget or widget will function. In so doing, we will be *defining our design problem* much more clearly and realistically. (We will see that this is where we start to think about what will emerge as the *product specifications*, the formal statements of the properties and functionalities that our design must have.)

3.1.1 Object attributes and lists of objectives

Imagine that we are members of a design team that is consulting for a company that makes both low- and high-precision tools (with corresponding prices). The company's management, seeking to penetrate a new market, has given the team a charter more specific than designing a "safe ladder," to wit, "Design a new ladder for electricians or other maintenance and construction professionals working on conventional job sites." This is a fairly "routine" design task, but to really understand the goals of this design, we still need to talk with management, some potential users, some of the company's marketing people, and some experts. We also need to conduct our own brainstorming sessions. We will get a better understanding of what our design project is really about by asking questions such as:

- What features or attributes would you like the ladder to have?
- What do you want this ladder to do?
- Are there already ladders on the market that have similar features?

And while asking these three questions, we will also want to ask:

- What does that mean?

As a result of our discussions and brainstorming, we might generate the list of characteristics and attributes of a safe ladder design shown in List 3.1

List 3.1 SAFE LADDER Attributes List

Ladder should be useful
Used to string conduit and wire in ceilings
Used to maintain and repair outlets in high places
Used to replace light bulbs and fixtures
Used outdoors on level ground
Used suspended from something in some cases
Used indoors on floors or other smooth surfaces
Could be a stepladder or short extension ladder
A folding ladder might work
A rope ladder would work, but not all the time
Should be reasonably stiff and comfortable for users
Step deflections should be less than 0.05 in
Should allow person of medium height to reach/work at levels up to 11 ft
Must support weight of an average worker
Must be safe
Must meet OSHA requirements
Must not conduct electricity
Could be made of wood or fiberglass, but not aluminum
Should be relatively inexpensive
Must be portable between job sites
Should be light
Must be durable
Needn't be attractive or stylish

We note in examining this list that not all of the statements are of the same kind. Some of them can be considered as binary issues, answered either by a "Yes" or a "No," while others allow for a range of answers. For example, the statement "Must not conduct electricity" doesn't really leave any options: either the ladder is a conductor or it is not a conductor, and in this case it must be an insulator. On the other hand, statements such as "Should be relatively inexpensive" allow for a range of prices at which the ladder could be built and sold. A ladder that can be built for $15 is more desirable than one that can be built for $20, assuming all the other characteristics are the same.

There are also other differences in these statements. The material of which the ladder is to be made (wood or fiberglass, but not aluminum) is a design choice that should probably be deferred until later in the design process, unless there is some specific information from the client that forces an early choice. The idea that the ladder must resist certain forces in certain ways (e.g., a limit on the deflection of a step) is a reflection of the way that engineers begin to translate features of a design into specifications (a subject we return to in

Chapter 5). The differences among the statements in List 3.1 are of considerable interest to the designer as they reflect differences between fundamentally different intellectual objects, a subject we address in Section 3.2.1.

3.1.2 Goals and objectives, constraints, functions, and implementations

The reason that the statements in the foregoing list for the safe ladder seem different in kind is that they reflect different intellectual objects that must be considered and evaluated by a designer. There are clearly some *objectives* we want to achieve (e.g., the ladder should be relatively inexpensive), some *constraints* (e.g., a step should deflect no more than 0.05 in.), some *functions* (e.g., to support workers), and there are some *means* or *implementations* (e.g., the ladder could be made of wood or fiberglass). We have already defined some of these terms in Section 1.2, but we will review them here with the intent of being able to recognize them in the context of an attribute list having different kinds of statements that would emerge very early in a design project, such as the safe ladder list just given.

Objectives are the desired attributes of a design.

Objectives or *goals* are ends that the design strives to achieve. (We generally view design objectives and design goals as meaning much the same thing, with the possible exception that reference to *the* design goal as the top-level objective.) Objectives are expressions of the desired attributes and behavior that the client or potential users would like to see in the designed object. They are normally expressed as "being" statements that say what the design will *be*, as opposed to what the design must *do*. For example, saying that the ladder should be portable is a "being" term. "Being" terms identify attributes that make the object "look good" in the eyes of the client or user, expressed in the natural languages of the client and of potential users.

Another way of identifying objectives is to note that they are often written as statements that "more (or less) of [the objective]" is better than "less (or more) of [the objective]." For example, lighter is better than heavier if our goal is portability. As such, objectives lend themselves to being measured somehow. In this way, objectives help us to choose among alternative design configurations.

Constraints are (strict) limits that a design must meet to be acceptable.

Constraints are restrictions or limitations on a behavior or a value or some other aspect of a designed object's performance. Constraints are typically stated as clearly defined limits whose satisfaction can be framed into a binary choice, for example, the ladder material is a conductor or it is not, or the step deflection is less than 0.05 in. or it is not. Constraints are important to the designer because they limit the size of a design space by forcing the exclusion of unacceptable alternatives. For example, any ladder design that fails to meet OSHA standards will be rejected.

Objectives and constraints sometimes seem to be interchangeable, but they are not. They are, however, closely related. Constraints limit the size of the design space, while objectives permit the exploration of the remainder of the design space. That is, constraints can be formulated to allow us to reject alternatives that are unacceptable, while objectives allow us to select among design alternatives that are at least acceptable, or, in other words, they *satisfice*. Designs that satisfice (or alternative selections in situations that require a choice to be made) may not be the best or optimal, but they do, at least, minimally satisfy all constraints. For example, we could barely satisfy OSHA standards or we could satisfy them "in spades" by making a "super safe" ladder in order to obtain a marketing advantage. Or, on the price side, a goal that a ladder should be "low cost" could be cast as a statement that the cost to build the ladder cannot exceed $25. In that case we have a fixed limit, a constraint. If we have *both* a low-cost objective and a $25 constraint, we may be able to exclude some initial designs based on the constraint alone, leaving us free to choose among the remaining designs based on cost and other, noneconomic objectives.

It is important to recall that both objectives and constraints *refer to the object being designed*, not to the design process. The "low-cost ladder," for example, has a low manufacturing or production cost. The cost of the design process (engineering salaries, market surveys, prototype development, etc.) may be high, but that's a separate matter altogether.

Functions are actions that a successful design must perform.

Functions are the things a design is supposed to *do*, the actions that it must perform. In our initial attributes list, functions are usually expressed as "doing" terms that often reflect the language of the engineer. We will discuss functions in greater detail in Chapter 4.

Implementations are specific choices of design options.

Lastly, *implementations* or *means* are ways of executing those functions that the design must perform. These are the items on the attributes list that provide specific suggestions about what a final design will look like or be made of (e.g., the ladder will be made of wood or of fiberglass), so they often appear as "being" terms. However, it is usually pretty obvious which "being" terms are goals to be achieved and which "being" terms are very specific properties. It is premature for us to consider means in great detail here, since any means or implementations that we might select would be governed by the things that a specific designed object must do. That is, implementations and means are very much *solution-dependent* in that they are often design choices made to implement the functions that are to be performed by the already-chosen design.

We can now pare the attributes List 3.1 by removing or pruning the constraints, functions, and implementations, leaving only objectives on the list. Thus, our pruned list of objectives for the ladder is given in List 3.2.

List 3.2 SAFE LADDER Pruned Objectives List

Ladder should be useful
Used to string conduit and wire in ceilings
Used to maintain and repair outlets in high places
Used to replace light bulbs and fixtures
Used outdoors on level ground
Used suspended from something in some cases
Used indoors on floors or other smooth surfaces
Should be reasonably stiff and comfortable for users
Should allow person of medium height to reach/work at levels up to 11 ft
Must be safe
Should be relatively inexpensive
Must be portable between job sites
Should be light
Must be durable

While List 3.2 is useful as a list of goals to be achieved, there is much more that we can do with it. In particular, if our list was much longer, we might find it difficult to use the list without organizing it in some way. We may want to group or *cluster* these objectives together in some coherent way. One way to start grouping entries on the list is to ask ourselves why we care about them. For example, why do we want our ladder to be used outdoors? The answer is probably because that's part of what makes a ladder useful, which is another entry on our list. Similarly, we could ask why we care whether the ladder is useful. In this case, the answer is not on the list—we want it to be useful so that people will buy it. Put another way, usefulness makes a ladder marketable. This suggests that we need an item on our list about marketing, for example, "The ladder should be marketable." This turns out to be a very helpful objective, since it tells us why we want the ladder to be cheap, portable, etc. If we go through clustering questioning of this sort, we will find a new list that we can put in the form of an *indented outline*, with *hierarchies* of major headings and various degrees of subheadings, as shown in List 3.3.

List 3.3 SAFE LADDER Indented Objectives List

0. *A safe ladder for electricians*

- **1. The ladder should be safe**
  - 1.1 The ladder should be stable
    - 1.1.1 Stable on floors and smooth surfaces
    - 1.1.2 Stable on relatively level ground
  - 1.2 The ladder should be reasonably stiff
- **2. The ladder should be marketable**
  - 2.1 The ladder should be useful
    - 2.2.1 The ladder should be useful indoors
      - 2.2.1.1 Useful to do electrical work
      - 2.2.1.2 Useful to do maintenance work
    - 2.2.2 The ladder should be useful outdoors
    - 2.2.3 The ladder should be of the right height
  - 2.2 The ladder should be relatively inexpensive
  - 2.3 The ladder should be portable
    - 2.3.1 The ladder should be light in weight
    - 2.3.2 The ladder should be small when ready for transport
  - 2.4 The ladder should be durable

As we can see, this revised, indented outline allows us to explore each of the top-level objectives further, in terms of the subobjectives that tell us how to realize it. At the highest level, our objectives turn us back to the original design statement we were given, namely to design a safe ladder that can be marketed to a particular group.

Now, we have certainly not exhausted all the questions we could ask about the ladder, but we can identify in this outline some of the answers to the three questions mentioned just

We have answered "How are you going to do that?" by identifying several subgoals or ways in which the ladder could be useful within the "The ladder should be useful" cluster and by specifying two further "sub-subgoals" about how the ladder would be useful indoors. And we have answered the question "Why do you want that?" by indicating that the ladder ought to be cheap and portable in order to reach its intended market of electricians and construction and maintenance specialists.

We can represent the indented outline we have just started in graphical form simply by laying out a *hierarchy* of boxes, each of which contains an objective for the object being designed, as shown in Figure 3.1. Each layer or row of objective boxes corresponds to a level of indentation (which is indicated by the number of digits to the right of the first decimal point) in the outline. Thus, the indented outline becomes an *objectives tree*: A graphical depiction of the objectives or *goals for the artifact* (as opposed to goals for a design project or process). The top-level goal in an objectives tree, which we represent as a node at the peak of the tree, is decomposed or broken down into subgoals that are at differing levels of importance or that include progressively more detail, so the tree reflects an *hierarchical structure* as it expands downward. An objectives tree also shows that related subgoals or similar ideas can be *clustered* together, which gives the tree some organizational strength and utility.

Objectives trees are ordered lists of the desired attributes of a design.

Still further, the graphical format of the tree is quite useful for discussions with clients and other participants in the design process. It is also useful for determining what things we need to measure, since we will use these objectives to decide among alternatives. The graphical format or tree is also useful since it corresponds to the mechanics of the process that many designers follow. Often, the most useful way of "getting your mind around" a large list of objectives is to put them all on Post-It™ notes, and then move them around until the design team is satisfied with the tree. We will discuss some of the mechanics of tree building and problem definition in Section 3.5.

The process just outlined—from lists to refined lists to outlines to trees—has a lot in common with one of the fundamental skills of writing, being able to construct an outline. A

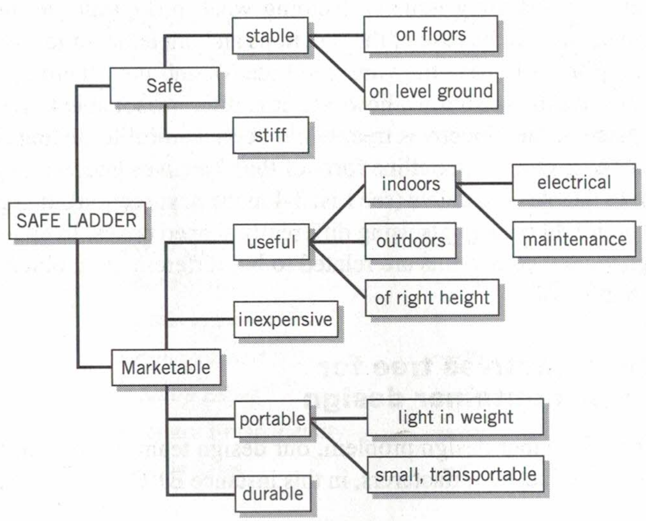

FIGURE 3.1 The objectives tree for the design of a safe ladder. It shows the first fruits of problem definition. Note the hierarchical structure and the clustering of similar ideas.

topical outline provides an indented list of topics to be covered, together with the details of the subtopics corresponding to each topic. Since each topic represents a goal for the material to be covered, the identification of an objectives tree with a topical (or an indented) outline seems logical.

We answer "how" by digging deeper into an objectives tree.

One final point about this simple example. Note that as we work *down* the tree, or move further in on the levels of indentation, we are doing more than just getting into more detail. We are also answering a generic *how* question for many aspects of the design, that is, the question of "*How* are you going to do that?"

We answer "why" by searching higher on an objectives tree.

Conversely, as we move *up* the tree, or further out toward fewer indentations, we are answering a generic *why* question about a specific (needed) function, that is, the question of "*Why* do you want that?" This enables us to track why we want some feature or other fine point in our design, which may be very important if we have to trade off features, one against the other, because the values of these features may be directly attributable to the importance of the goals they are intended to serve. We will say more about this in Section 3.3.

3.1.3 How deep is an objectives tree? What about pruned entries?

Where do we end our list or tree of objectives? The simple answer is to stop when we run out of objectives or goals and implementations begin to appear. That is, within any given cluster, we could continue to parse or decompose our subgoals until we are unable to express succeeding levels as further subgoals. The argument for this approach is that it points the objectives tree toward a *solution-independent* statement of the design problem. That is, we know what characteristics the design has to exhibit without having to make any judgment about how it might get to be that way. In other words, we determine the attributes of the designed object without specifying the way the objective is realized in concrete form.

Another way of limiting the depth of an objectives tree is to look out for verbs or "doing" words because they normally suggest functions. Functions should definitely not appear on objectives trees or lists.

A second tree-building issue is deciding what to do with the things that we have removed from the list. In the case of the functions and implementation, we simply put them aside (recording them in case they are good ideas), and pick them up again later in the process. In the case of constraints, however, it is often reasonable to reenter them into an appropriate place in the objectives tree, while being careful to distinguish them from the objectives. For example, in an outline form of the objectives tree, we might use italics or a different font to denote constraints (see List 3.4 in the next section). In a graphical form, we may wish to highlight constraints using differently shaped boxes. In either case, it is important to recognize that constraints are related to but different from objectives, and they are used in different ways.

3.1.4 The objectives tree for the beverage container design

In the beverage container design problem, our design team is working for one of the two competing food product manufacturers, in this instance BJIC. (We note parenthetically that there is an interesting ethical problem that we will address in Chapter 10, that is, could our design team, or our firm, take on the same or similar design tasks for both, or for two competing clients?) However, for now, let us suppose that we're dealing with a single client and that our client's project statement is as stated in Section 3.5: "Design a bottle for our new juice product."

In order to clarify what was wanted from this design, our design team questioned many people in BJIC, including the marketing staff, and we talked to some of their potential customers or users. As a result, we found that there were several motivations driving the desire for a new "juice bottle," including: plastic bottles and containers all look alike; the client, as a national producer, has to deliver the product to diverse climates and environments; safety is a big issue for parents whose children might drink the juice; many customers, but especially parents, are concerned about environmental issues; the market is very competitive; parents (and teachers) want children to be able to get their own drinks; and, finally, children always spill drinks.

These motivations emerged during the questioning process, and their effects are displayed in the augmented attributes list for the container given as List 3.4. Some of the entries in this list are shown in italics because they are constraints. Thus, these constraint entries can be removed from a final list of the attributes that are objectives (to be reinserted later, as discussed above).

List 3.4 BEVERAGE CONTAINER Augmented Attributes List

Safe	→	DIRECTLY IMPORTANT
Perceived as Safe	→	Appeals to Parents
Inexpensive to Produce	→	Permits Marketing Flexibility
Permits Marketing Flexibility	→	Promotes Sales
Chemically Inert	→	*Constraint* on Safe
Distinctive Appearance	→	Generates Brand Identity
Environmentally Benign	→	Safe
Environmentally Benign	→	Appeals to Parents
Preserves Taste	→	Promotes Sales
Easy for Kids to Use	→	Appeals to Parents
Resists Range of Temperatures	→	Durable for Shipment
Resists Forces and Shocks	→	Durable for Shipment
Easy to Distribute	→	Promotes Sales
Durable for Shipment	→	Easy to Distribute
Easy to Open	→	Easy for Kids to Use
Hard to Spill	→	Easy for Kids to Use
Appeals to Parents	→	Promotes Sales
Chemically Inert	→	*Constraint* on Preserves Taste
No Sharp Edges	→	*Constraint* on Safe
Generates Brand Identity	→	Promotes Sales
Promote Sales	→	DIRECTLY IMPORTANT

The augmented List 3.4 also shows how, after additional brainstorming and questioning, some of the listed goals are either expanded into subobjectives (or subgoals) and others are connected to existing goals at higher levels. In one case a brand new top-level goal, Promote Sales, is identified. The objectives tree corresponding to (and expanded from) this augmented attribute list is shown in Figure 3.2, and a tree combining objectives and constraints is shown in Figure 3.3. The detailed subgoals that emerge in these trees clearly track well with the concerns and motivations identified in the clarification process.

As a result of the thought and effort that went into List 3.4 and the objectives trees of Figures 3.2 and 3.3, the design team rewrote and revised the problem statement for this design project to read: "Design a safe method of packaging and distributing our new children's juice product that preserves the taste and establishes brand identity to promote sales to middle-income parents." Thus, as we noted in Chapter 2, one of the outputs of the design preprocessing (or problem definition) phase is a revised statement that reflects what has been learned about the goals for a design project. That is, the emergence of a clearer understanding of the client's design problem results in an objectives tree that points toward the expression of the features and behaviors wanted from the designed object, and it often results in a simultaneous revision or restatement of the client's original problem statement.

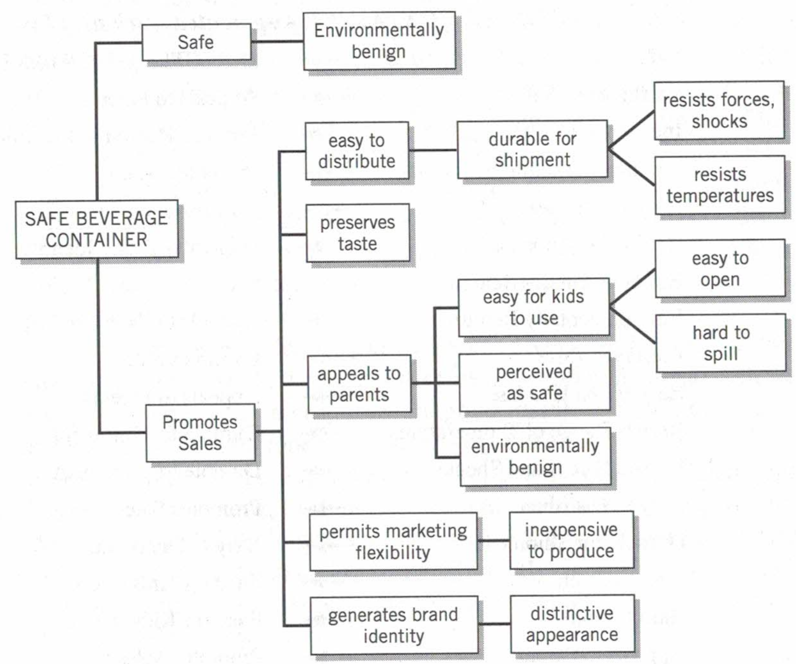

FIGURE 3.2 The objectives tree for the design of a new beverage container. Here the work on problem definition has lead to a hierarchical structuring of the needs identified by the beverage company and by the potential consumers—or at least the consumers' parents!—of the new children's juice drink.

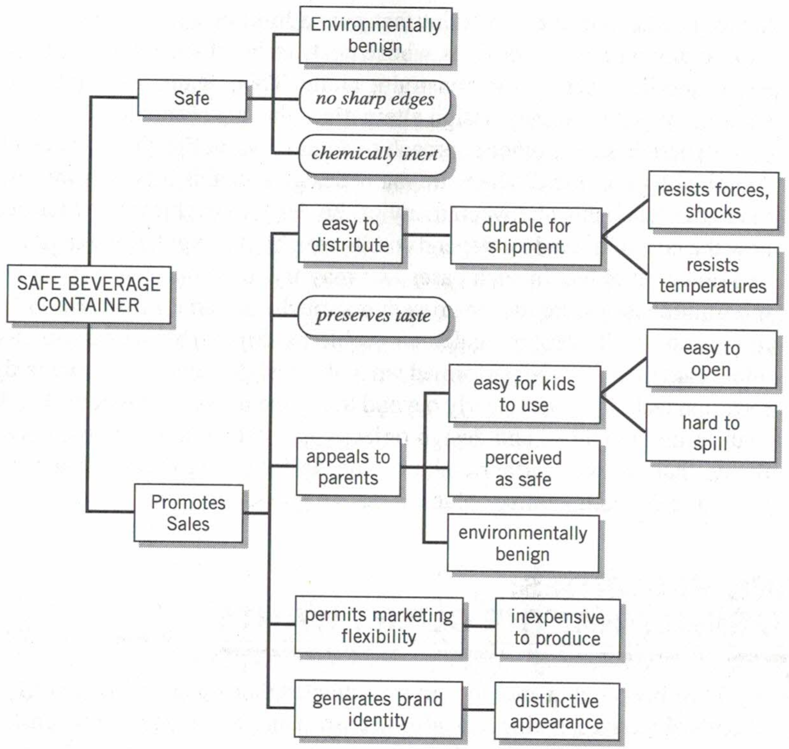

FIGURE 3.3 A combined tree (objectives in rectangles and constraints in ovals) for the design of a new beverage container. Here the goals for the new product are shown together with the constraints that apply to the object being designed.

3.2 CONSTRAINTS: SETTING LIMITS ON WHAT THE CLIENT CAN HAVE

There are limits to everything. That is why constraints are extremely important in engineering design, as we noted in Section 3.1.2 when we articulated some differences between constraints and objectives.

Constraints enable us to identify and exclude unacceptable designs.

As a practical matter, many designers use constraints as a sort of "checklist" for designs in order to prune the set of designs to a more manageable size. Such constraints, which can be included in properly marked trees that include both objectives and constraints, are usually expressed in terms of specific numbers. By way of contrast, objectives are normally expressed as verbal statements that sometimes can be formulated in terms of continuous variables or numbers that may allow for a range of values of interest to the designer. To reiterate our earlier illustration of this point, a goal that a ladder should be cheap could be stated in terms of having a materials or manufacturing cost that does not exceed a fixed limit or constraint, say \$25. On the other hand, we could have *both* an objective that the ladder be cheap *and* a constraint that puts a limit on the cost. In that case we are able to choose among a set of designs whose costs to build are different, as long as they are all below the limit set by the constraint alone. This, again, is the strategy of "satisficing" wherein we select among design alternatives that are acceptable.

There is still another approach to dealing with objectives that can be cast in terms of "continuous variables." There are many design domains in which we can formulate mathematical relationships between many of the design variables. For example, we might know how the cost of the ladder depends on its weight, its height, the size of its projected market, and other variables. In such cases, we may try to *optimize* or get the best design, say the minimum cost ladder, using procedures much the same as we use to find the maxima or minima of multi-variable calculus problems. Similarly, *operations research* techniques allow calculations to be performed when the design variables are explicitly numerical. Optimization techniques are clearly beyond the scope of our discussions, but the underlying idea that design variables, and design objectives, interact and vary with one another, is also a theme that we will further elaborate in the following section when we discuss ways of assessing the comparative values of design goals.

3.3 SETTING PRIORITIES: RANK ORDERING WHAT THE CLIENT WANTS

We have been rather insistent in this chapter that we properly identify and list all of the client's objectives, taking care also to ensure that we don't mix up constraints, functions, or means with the goals set for the object being designed. But do we know that all of the identified objectives have the same import or value to the client or the users? So far we have assumed that each of the top-level objectives has the same value to all concerned because we have made no effort to see whether there is any variation in perceived value. It is almost certain that some goals will be more important than others, so we ought to be able to recognize that and measure it. How are we going to do that?

3.3.1 On measuring things

Let us step back for a moment and think about what it means to measure and compare some design objects, whether they be the objectives for a design, a set of design attributes, or a collection of conceptual designs. It is not obvious that there is any way to meaningfully mark or "plot" the individual design objects along an axis, but it does makes sense to plot evaluation points earned by each object along a line. Those evaluation points can be compared and some design decisions can be taken. How do we award such evaluation points? Is there a scale we can use to lay out the evaluation points for each design?

Engineers are used to measuring all sorts of things: beam lengths, surface areas, hole diameters, speeds, temperatures, pressures, and so on. In each of these cases, there is a ruler involved that shows a zero and has marks that show units, whether they be inches, microns, mm of Mercury, or degrees Farenheit or Centigrade. This establishes a common basis for comparing. Without rulers, how would we meaningfully quantify the assertion that "A is taller than B"? Simply standing A and B against each other, back to back, doesn't cut it (especially if A and B are not easily moved). However, by using a measuring stick that has