Institute of Design
ILLINOIS INSTITUTE OF TECHNOLOGY

Covering User Needs

Charles L. Owen

Founded as the New Bauhaus in 1937, Chicago's Institute of Design, IIT
is a center for advanced study in human-centered innovation.
Learn more: www.id.iit.edu

November, 2007

A web article written for:

The Business Process Management Institute
Web site: **www.BPMInstitute.org**
Main Section: ***Articles***
or
Section: ***Innovation;*** Topic: ***Papers;*** Subtopic: ***Articles***
November 29, 2007

Keywords: innovation, planning, analysis, problem coverage, design

Previous articles in the series:

Design Thinking: Driving Innovation*, September 2006*
Another Look at Quality*, December 2006*
First Things First*, February 2007*
Reforming the Development Process*, May 2007*
Goals and Definition*, July 2007*

Abstract

The problem of analysis in system planning has two parts: insight and coverage. Both have to do with understanding who the users of a system are and what they do with the system.

The problem for insight is discovering what goes wrong (or right) as users work with the system. Information elements associated with insight are qualitative and the source of ideas likely to be creative and supportive of systemic solutions. They are discussed in detail in another article.

The problem of coverage, the main topic of this article, is enumerating what the system must do to meet the needs and expectations of its users. A useful framework for doing this is a Function Structure. Function Structures are derived with familiar top-down analysis techniques that establish, first, major Modes of behavior, then the Activities undertaken or to be undertaken within these Modes, and finally the Functions that compose or will compose the Activities. Functions are information elements at an action level—ideal for establishing system coverage and valuable for evaluating system performance.

Two problems frequently encountered in planning for new product development are what I call the "depth problem" and the "breadth problem". In my article "First Things First", I talked about the depth problem—basically failing to spend time and resources establishing what to make or implement (the *right* concept) before committing to planning how to make it. In this article, I'll focus on the breadth problem—failing to consider the full range of users, and its remedy: establishing who the users are and the aspects of system functionality they are concerned with.

Notice I just said "system". I'll talk more about that in a later article, but it is worth mentioning that just about any subject for planning can be usefully viewed as a system in the general sense that multiple components exist, they come in a variety of forms from hardware to procedures, communications, events—even policies—and the relations among them are as fruitful subjects for planning as the components themselves.

The Basic Analysis Problem

As Structured Planning has taken form as a planning process, an important problem has been the characterization of information elements, the fundamental building blocks of analysis. Early on, I and others thought the best representation might be some form of *requirements* that needed to be satisfied. This turned out to be too prescriptive for truly creative planning, and we moved on to a more sophisticated, neutral model we called *observations*—observations of human and/or system behavior. The premise we were working with was that information elements ought to be qualitative and insightful. Such information would provide understanding in depth, and that would trigger innovative system solutions.

Variations on the theme improved the amount of information that these "elements" contained. One conceptual breakthrough was an information format we called "Force-Tendency Statements", first conceived at the University of California Berkeley. Force-Tendency Statements are two-clause statements of insight expressing a cause/effect relationship (if that can be substantiated) or, more likely, a weaker relationship that, although less conclusive, still spotlights a condition that can create a problem and should be addressed.

A simple example of this kind of statement (from a marine ecosystems project) was: *If two people use one flashlight for night collecting, only the person who controls it can see clearly what is in the lighted area.* From experience, anyone recognizes the truth of this; the insight comes when we realize that it is a problem of accommodation as the eyes of a person following along without the light move in and out of the lighted area. The person controlling the light has no accommodation problem because hand/eye coordination keeps his or her eyes in the pool of light.

The quality of information carried in these kinds of elements is significant and strongly supportive of creative solution, but a problem still remains: coverage. The analysis problem is really two-part; you need insights, but you also need assurance that you are reasonably covering the full range of the problem context. The characterizations we had developed gave us deep insights about some aspects of a problem, but no information where we had no insights. What we said at the time was, "The rest (what we have no insights for) are routine problems that will be covered in the normal course of development by a competent designer"... but that was not sufficient.

Functions and Function Structures

The answer was to separate insight and coverage. If insight is about how and why things go wrong (or right) as actions proceed, coverage is about discovery of the full range of users, delineation of the activities they will be expected to engage in, and enumeration of the actions expected to take place in these activities as they work with the system. Insight is a very important topic for innovation, and I will discuss that at greater length in another paper, but coverage is also critical, and I deal with that here.

The first step in establishing coverage is recognizing that everything planned or designed exists in time. Even simple things like ashtrays and non-physical concepts such as organizations or rule sets such as constitutions can be usefully thought about from the perspective of life cycle. And the atomic form of elements best associated with time is an action—or Function. I prefer Function as the elemental term because functions are more easily seen as having purpose, and system design is all about purpose.

The goal of the coverage side of analysis is to establish all the functions that the system has to perform or the user has to perform with it (or as many as reasonable, given time and human resources). This is most easily done by top-down analysis, a process almost everyone uses naturally in the course of every-day activities. The process begins at a relatively-abstract high level and proceeds hierarchically to a level where Functions can be expressed succinctly as specific actions. The result is a Function Structure, a tree structure topped by a single-node system title surmounting layers of category nodes increasing in number and specificity to a base layer of Functions (see Figure 1).

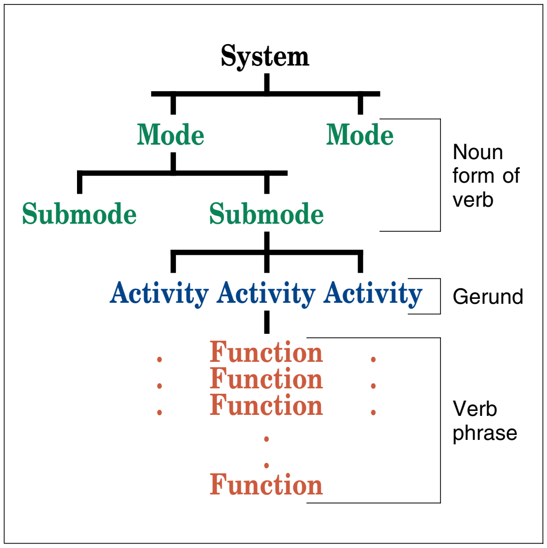

Figure 1 **Function Structure: The Three Analytical Layers**

Experience has shown that Function Structures can be generated most efficiently if the layers of nodes have descriptive properties appropriate to their level in the structure. Three separate forms—Modes, Activities and Functions—are sufficient.

At the highest level, nodes are best described as *Modes* of behavior or *Modes* of operation. As do all nodes in the structure, Modes express action; the structure is about functionality. But in the case of Modes, the action is process on a broadly inclusive scale. For an office furniture system, as an example, possible Modes might be Specification, Transport, Installation, Operation, Maintenance, Adaptation and Retirement. All are nouns created from verbs. All are "big" action categories appropriate to a top-level, initial breakdown of system functionality. As necessary, Modes are subdivided into Submodes with the same descriptive properties when more detail is useful.

At the middle level, the character of nodes changes to one more tightly time bound. Nodes here are *Activities*. I think of them as "purposeful performances"—highly analogous to scenes in a play; users in various roles are the actors, components of the system are the props they work with, and the environment in which the Activity takes place is the set. An activity can easily be thought of as having characteristic beginning and ending actions, goal-oriented actions in the middle, and some special actions that take place under unusual circumstances. To distinguish Activities from Modes, Activities are named with the gerund form of verbs, turning a verb into a

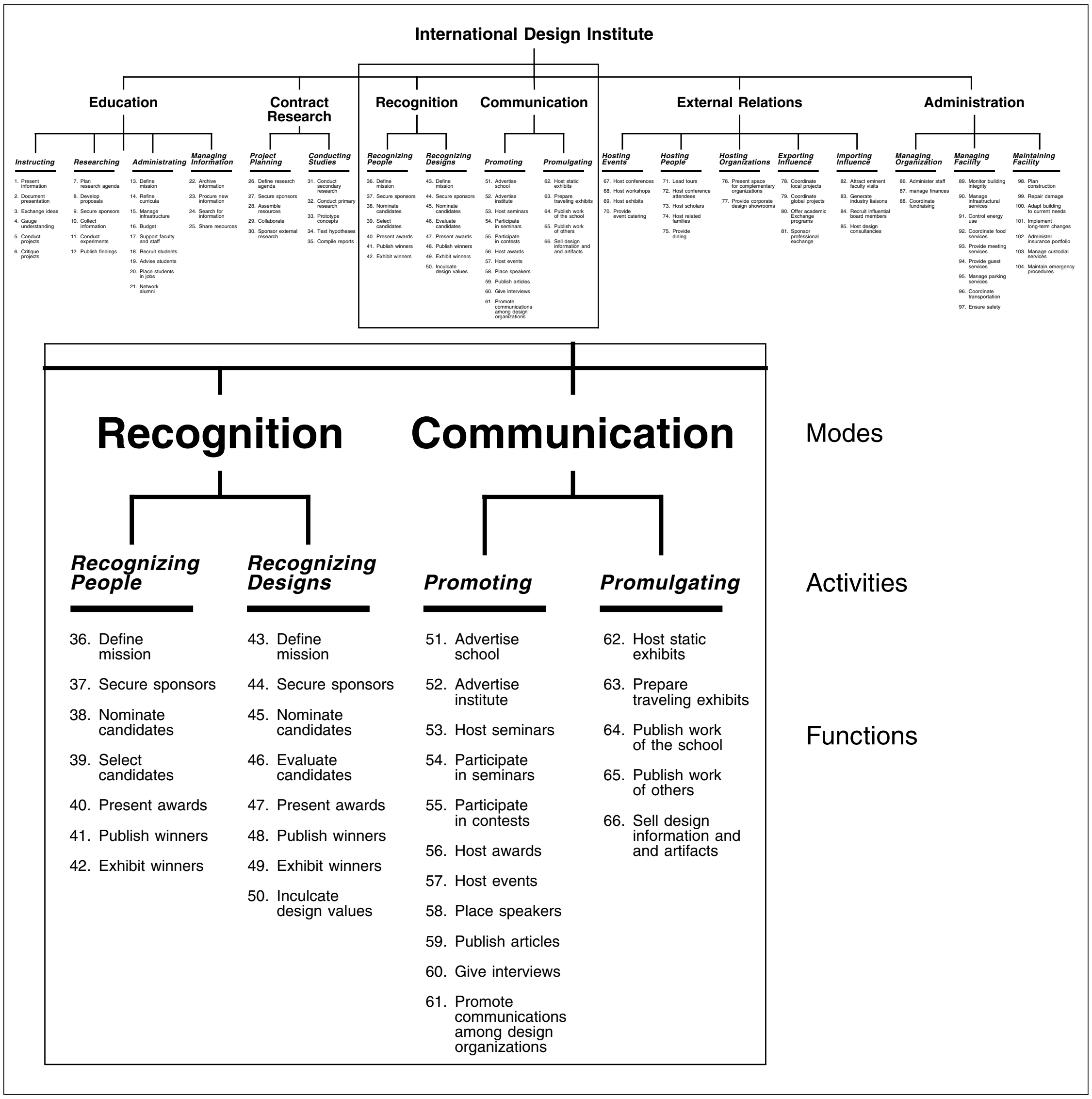

Figure 2 Function Structure for a Projected International Design Institute, an example of a relatively small framework for an institution design.

noun by ending it with -ing. Often, the same basic word root can be moved to different levels by simply changing its form. For example, Maintenance would be a Mode, but Maintaining would be an Activity. Specification would be a Mode; Specifying, an Activity.

Functions are at the lowest, primary level. As the most specific action elements, they are represented with verb phrases, the best means to express succinctly something that a system must do or a user must do with it. Verb phrases have an immediacy perfectly appropriate to the change of scale needed in moving from an Activity to its component actions. Again, by changing form, a verb root can be moved up and down the hierarchy: "Maintain open walkways in winter" is a Function; "Maintaining Grounds" is an Activity; and "Environmental Maintenance" is a Mode.

Usually five to ten Functions are sufficient to specify what should take place in an Activity. And usually 100 to 250 Functions are adequate for the Activities of an entire project (Figure 2). Controlling size is both possible and desirable. It is desirable in that this size is within the capabilities of a planning team of 4-5 members with 4-6 part-time months to complete the project. It is possible because the English language allows us relatively easily to find appropriate words to define Functions at higher and lower levels of generality, thus allowing us to cover an Activity appropriately with either more or less Functions. The number of words in Merriam-Webster's 3rd International Dictionary is about 450,000. Estimates suggest that there may actually be as many as a million words in English—far more than available in any other language.

Functions as Criteria

Once completed, a Function Structure is an organized set of criteria. But it and its Functions serve multiple purposes. As a tool for analysis, the Function Structure provides the framework for assembling action descriptions for what an intended system must do. In the process of synthesis, its Functions are reorganized into an Information Structure especially constructed to optimize the association of Functions for innovative concept building. And for evaluation, both Functions and Function Structure become the criteria against which the success of the system solution can be tested.

A good *system* will perform all the functions well for which it has been planned. A good *system concept* will have sought out and defined all the functions necessary to cover users' needs. Good system *coverage* will ensure that all system users have their interests considered—not just customers and end users, but the many users that in any way affect or are affected by the system. For products as well as services, this means all involved in the acts of production, sales, specification, transport, operation, maintenance, repair, adaptation, retirement, etc. Establishing the Functions to be performed establishes the criteria to be met—a good system performs all its required functions well. Success confers product integrity.