Structured Planning

Introduction

Structured Planning is a process for finding, structuring, using and communicating the information necessary for design and planning activities. It is a front-end process for developing concepts.

A number of projects have been undertaken with it and used to continue its development. Among well over 60 of these, an early published project for Chicago's transit authority (CTA) was ***Getting Around: Making the City Accessible to Its Residents*** (1972). In 1983, the ***House of the Future*** project won the Grand Prize in the **Japan Design Foundation**'s First International Design Competition. In 1985, a project on ***Space Station*** was undertaken for **NASA**; in 1987, the ***Aquatecture*** project again won the Grand Prize in the **Japan Design Foundation**'s Third International Design Competition. In 1991 ***Project Phoenix*** on global warming was honored as Environmental Category Grand Winner in **Popular Science** magazine's "100 Greatest Achievements in Science and Technology" for the year. In 1993, two projects, ***NanoPlastics*** and ***Aerotecture***, won awards and were widely publicized in Europe and Japan, and in 1995 the ***National Parks*** project developed plans for the future of the National Park Service. As the process has evolved, it has become an increasingly useful planning tool for products, systems, services and organizations. It is now being used commercially.

This document provides a general overview of Structured Planning using a 2001 project for the National Center for State Courts: ***Access to Justice: Meeting the Needs of Self-Represented Litigants.***

Defining a Project

Projects cannot be prescribed absolutely. There is always something more to say about issues that should be addressed. Nevertheless, it is important to take stands on how a project should proceed in the early stages of specification. These stands, or positions, are formative and help to clarify issues and limitations that must be recognized, as well as special viewpoints that exist within the planning team.

The Structured Planning process begins with a **Charter**. This is a "brief" that sets out what must be done without overly burdening the project with preconceived ideas or conceptual frameworks.

Access to Justice

Develop integrated concepts for improving access to justice for those who choose or are forced to represent themselves in court.

Using Structured Planning methodology, conduct an advanced planning project to develop concepts for an integrated system solution. The proposed solution should be sustainable, scalable and adaptable to changing needs.

Figure 1. A **Project Statement** *is a succinct sentence that describes the goal of the project in operational rather than noun-name terms.*

The Charter serves as an initial communication vehicle between client and planners. It contains background, context, basic goals and a project statement that cuts to the heart of the planning task (Figure 1). Definition then builds around these foundation materials and project statement with the addition of "white papers" on issues that must be addressed. In the Structured Planning process, these are called **Defining Statements.**

Defining Statements serve to focus the project within the general direction of the project statement. They pick out issues that are important and suggest the specific direction that the project should follow with regard to them. The word *issue* is used advisedly with the intention that the subjects for Defining Statements should be particularly selected from topics that are controversial, or at least have plausible alternatives associated with them. Figure 2 shows two Defining Statement examples.

To make it easier for team members to cooperate in the generation of Defining Statements, they are carefully written to a common format. The format is five-part: (1) ***Issue Topic***—one or two words establishing the subject of the Defining Statement; (2) ***Question at Issue***—a short question raising an important issue under the topic; (3) ***Position***—a sentence stating the position to be taken on the issue, (4) ***Alternative Positions*** —other plausible positions that were considered, but not taken; and (5) ***Background and Arguments*** —as much discussion as is necessary (in narrative form) to explain the reason/s why the position was selected (and why others were not). There are three kinds of Defining Statements, differentiated by the force they exert on the planning process.

Defining Statement | Issue Topic: **Legal Procedure** | **11**

Project
Access to Justice

Originator
Charles Owen

Contributors

Source/s
Discussion with Judge Roger Warren, President, National Center for State Courts, at the Institute of Design, 30 March, 2001.

Question at Issue
How far should the system go in suggesting changes to legal procedure?

Position
[x] **Constraint** [ ] **Objective** [ ] **Directive**
The system must deal fundamentally with both legal procedures and supporting information systems, managing reform through the introduction of better decision support and information processes.

Alternative Positions
[x] **Constraint** [ ] **Objective** [ ] **Directive**
The system must concentrate on basic reforms to procedure, taking risks as necessary to incorporate innovative simplification.

[x] **Constraint** [ ] **Objective** [ ] **Directive**
The system must concentrate on improvements to information and communication processes, avoiding risky changes to legal procedure that might prove unacceptable.

Background and Arguments

Civil court procedures have evolved in complexity to the extent that, today, rather than protecting rights and guaranteeing fairness, they often actually impede the effective administration of justice. Layers of formal procedures (that serve the interests of lawyers more than those of either court or litigants) take up precious time, add confusion and put litigants at serious disadvantage in navigating a labyrinthine legal process. The result is frustrating, costly—and largely unnecessary.

The problem for reform is to simplify processes for all users while staying within the spirit and principles of the law. Three approaches merit consideration.

The first is to revise court procedures fundamentally. In this approach, the process is abstracted, expected outcomes are established, typical initial conditions are enumerated, and the most direct means to ascertain condition and assign outcome are then incorporated in procedure. All classical court procedures are open to question and revision or dismissal.

The second approach leaves court procedures basically intact, but regards all requests for information and processes for obtaining it as subject to reform. The overall process is treated as communication with the approach being to simplify requirements for information and how it is generated.

The third approach takes a path between procedural reform and communication reform. In this model, procedures are augmented or supplemented with processes to improve decision making *and* information processing.

The first model has strong appeal because it streamlines access to justice at a fundamental level—the reform of court procedure. It has three difficulties: first, it is potentially politically charged, running the risk of unacceptability; second, it is sensitive to legal error in that statutes must not be violated; and third, it ignores benefits available at no risk from the application of information technology. The second model is attractive for its safe approach to the legal issues, but falls short of the level of reform possible and desirable. It is too safe.

The third model has the greatest potential. It offers an integrated approach with the best of both of the other models. It offers more than either alone, while avoiding the risk of unacceptability and the disappointment of less-than-expected results.

Version **1** Date: 31 March, 2001 Date of first version: 31 March, 2001 Form: 5/19/1998

Defining Statement | Issue Topic: **Enforcement Assistance** | **15**

Project
Access to Justice

Originator
Jennifer Joos

Contributors
5 Feb., 2001 Holly Roeske
14 Feb., 2001 Ronald Staudt
26 Apr., 2001 Charles Owen

Source/s
Handbook on Child Support Enforcement. U.S. Department of Health and Human Services Administration for Children and Families. <http://www. pueblo.gsa .gov/cic_text/children/childenf/index.htm>

Zorza, Richard. **Designing from the Ground Up, A Self-Help Centered Court.** Washington, DC: State Justice Institute, In preparation.

Heller, Paul. *Speech on Collection.* Chicago, 2000.

Question at Issue
What part should the system play in assisting judgment creditors in the enforcement process?

Position
[ ] **Constraint** [x] **Objective** [ ] **Directive**
The system should support flexible procedures that streamline the process of enforcement and help successful litigants to enforce creditor judgments.

Alternative Positions
[ ] **Constraint** [ ] **Objective** [x] **Directive**
Programs and procedures ought to be established at federal and state levels to enforce the collection of judgments in civil cases.

[ ] **Constraint** [x] **Objective** [ ] **Directive**
Litigants should be wholly and independently responsible for collecting their judgments.

Background and Arguments

For the self-represented litigant, receiving a verdict is not the end of the problem; in fact, it only begins the harrowing process of enforcement. The procedures that constitute enforcement basically reprise the entire litigation process. Diagnosis, preparation and hearing must be repeated in order to achieve reparation. Collection is further complicated when the debtor refuses to cooperate and the litgant—with a judgment—is unable to locate either debtor or assets.

In Cook County, Illinois, supplemental proceedings currently require that unless the debtor (defendent) provides financial statements, the plaintiff must first file a Citation to Discover Assets, compelling the debtor to appear in court with proof of his or her assets. If the respondent fails to appear, a Rule to Show Cause is filed compelling the respondent to appear in court and explain why he was not in court the first time. If the respondent fails to appear again, an Attachment Order of the Court directs the sheriff to physically apprehend the debtor and bring him to court (Heller 2000). Clearly, the pursuit of collection can be a long process, potentially frustrating enough to lead a litigant to abandon a rightful claim. There is no consistent or comprehensive support system in place to aid litigants in their search for information about debtors and assets.

In 1975, the US Department of Health and Human Services established the Child Support Enforcement Program. State child support programs, on a local level, establish and enforce support orders and collect child support payments (Handbook on Child Support Enforcement). Information available to child support enforcement agencies could be helpful to anyone needing assistance enforcing a judgment. Child support enforcement programs are able to access information such as state tax files, real and titled personal property records, occupational and professional licenses and business information, information from employment security agencies, public assistance agencies, motor vehicle departments, and law enforcement departments as well as records of private entities such as public utilities and cable television companies. This includes names and addresses of individuals and their employers as they appear in customer records; information obtainable from financial institutions can include asset and liability data (Handbook on Child Support Enforcement).

Guidelines that standardize access to this wealth of information coupled with procedures to aid in the discovery of assets would ensure more successful enforcement of verdicts.

Version **4** Date: 26 April, 2001 Date of first version: 2 February, 2001 Form: 5/19/1998

Figure 2. A **Defining Statement** *helps to bound the problem. In decreasing strength from* **Constraint,** *to* **Objective** *and* **Directive,** *it sets goals for where good fit must be sought for the concept. These two are examples of a Constraint and an Objective.*

Constraints are the strongest statements. They state what must or must not be done. They fix positions that must be held as conscientiously as possible. The word **must** is used in the position statement to amplify the force of commitment.

Objectives are Defining Statements less forceful than Constraints, and more forgiving in their demands. It is possible to settle for less than complete satisfaction of an Objective, although the planning team will strive to achieve as much of its prescription as possible. The word **should,** which carries with it a sense of obligation, is appropriate for the position statement. In choosing between the Constraint or Objective labels for a Defining Statement, the decision is made with regard to the force of commitment that can reasonably be expected. If achievement cannot really be guaranteed, the statement probably should be an Objective. Objectives can be thought of as having more of a scalable measure of achievement than Constraints, which tend to be thought of as thresholds that must be observed.

Directives are somewhat different from the other two statement types. In the hierarchy, they have the least force and, accordingly, are used for goals that are *desirable*. They are also used to express the biases of the planning team. Everyone brings biases of style or preference to the projects they work on. Some planners become well enough known for them that they are sought out for the very *brand* or trademark their style places on a project. Unfortunately, all biases are not readily observable, but that doesn't mean that they should not be expressed! A major problem that often develops in client/planner relations stems from the failure of one or both parties to communicate the subtleties of their intent. The Directive provides a place for this kind of expressive statement. English also has a nice wording for this level of commitment: **ought to.** The words suggest almost a moral or ethical force—appropriate for a bias or a statement of style.

Developing Information

All things exist in time. They are not unchanging, and they cannot be designed without regard for the way they operate and are used over time. Any product can be viewed as a system operating with a user or users in different ways that are appropriate for its modes of existence. To plan effectively, a planning team must recognize these ***Modes,*** identify ***Activities*** that occur in them, and isolate the ***Functions*** that the system must perform (or the user must perform for it) within each Activity (Figure 3).

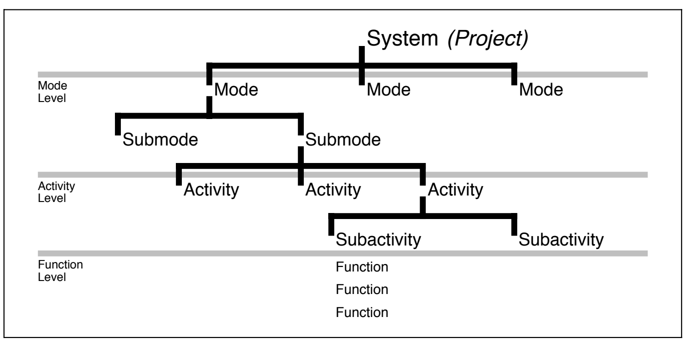


Figure 3. For thorough coverage of what must be considered in the design of a system, it is helpful to organize the analysis hierarchically. A three-level model enables the analyst to break down system actions nicely to find the primary **Functions** *it must perform through all its* **Modes** *of behavior.*

Typical Modes through which familiar hardware systems pass include: manufacture, distribution, transportation, storage, use, maintenance, repair, and retirement. For any given system, these may be replaced, augmented or supplemented with others; and major Modes may be subdivided into Submodes specialized for the individual case. In this project, the Modes were Diagnosis, Preparation to Initiate Proceedings, Alternative Dispute Resolution (ADR), Hearing, and Enforcement. Listing the Modes is generally not difficult, and the stage then is set to identify Activities that take place within them.

By definition, an ***Activity*** is a set of purposeful actions taken by users and system in an environmental setting. The actions of an Activity, thus, should be cohesive enough in purpose to be thought about collectively. Two difficulties make it hard to assign titles to Activities. First, the general complexity of real-life systems tends to make it difficult to bound Activities neatly. Second, the multiplicity of word choices available makes it difficult to find the right set of titles to achieve an intellectually satisfying balance. By trial and error, however, it is usually possible to name a set of Activities satisfactorily to cover the actions of a Mode neatly.

As a way to begin an analysis, it is helpful to think of Activities as scenes in a play. The analogy is completed by thinking of the set on which the play takes place as having props that are actively used in scenes (the system components) and others which provide background (environmental components). From scene to scene, new props may move into the center of attention, while ones of previous interest become background. Users, in the analogy, are the actors. The roles they assume reveal the special characteristics of users' interests.

Setting the stage for an Activity and playing out the scene enable the planning team to see the **Functions** that are involved in the "performance". It is these that must be identified, since these are, ultimately, what the system must do well (or help the user to do well). Each Activity entails the performance of a number of Functions, either by the system or by its users. Whether these Functions are retained in their original user or system categories in the final design is unimportant; Functions can be assigned and reassigned fluidly between user and system to obtain the best resolution of the problem within the set of Defining Statements. What is important is that a good coverage of the Functions is obtained.

Half of the purpose of the foregoing process is the enumeration of Functions. The other half is the development of information about these Functions that will shed insight on what happens as they are performed.

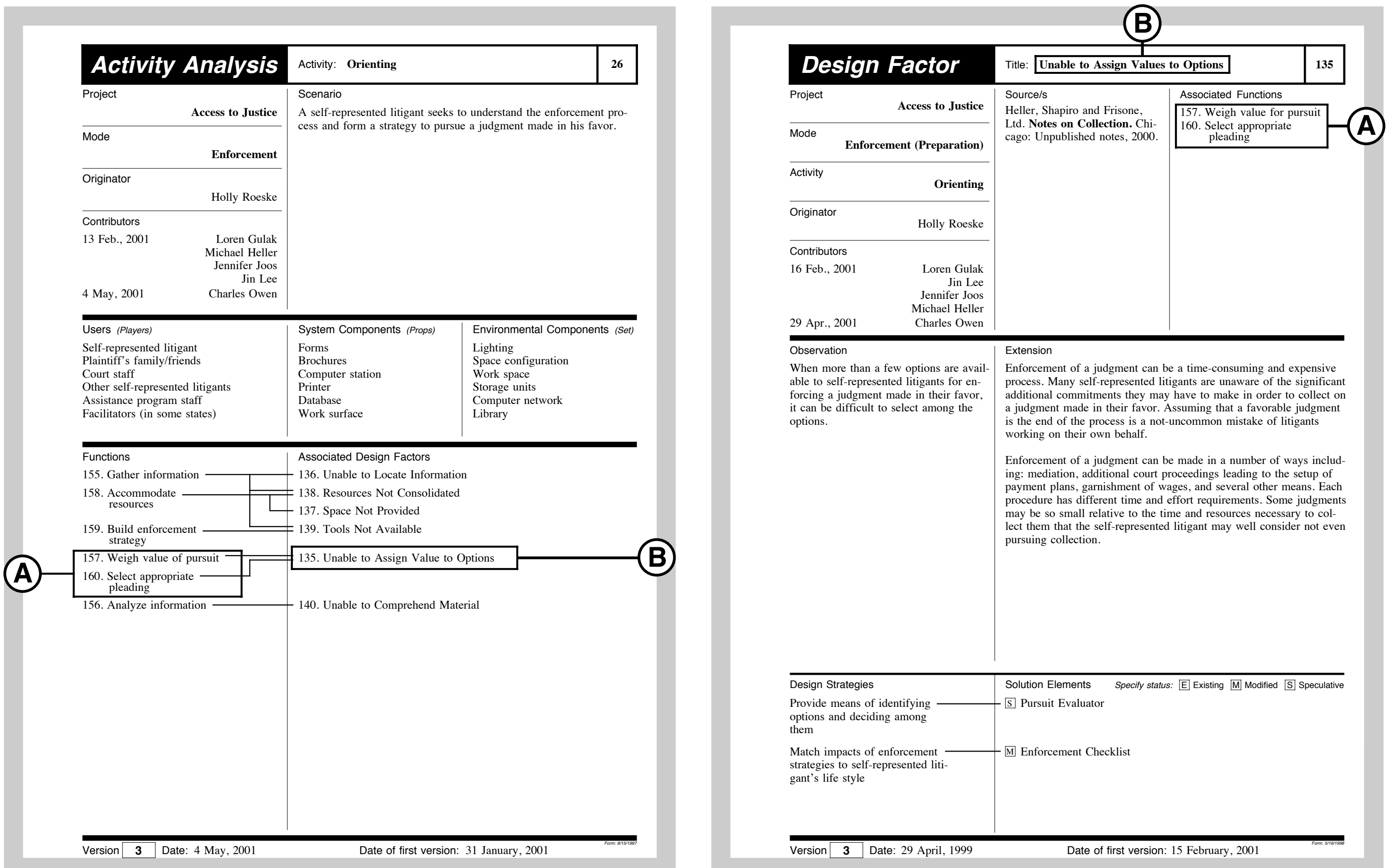
Treating the system to be designed as a user/system model allows it to be analyzed from the perspective of the system or of the user. From the system standpoint, classic systems analysis observes operations and determines relationships among components—toward the creation of a system model with features that can be described and processes that can be simulated. The analysis of Activities scrutinizes users' actions for the purpose of building an organization of Activities describing user behavior. Both kinds of analysis are useful for producing *hard* data and constructing a model. In fact, the process model just discussed draws from both. But the *hard* data is not enough to guarantee a good conceptual design.

What is necessary is *insight; information* as distinguished from *data*. Information has surprise—it reveals something not known before, or not thought of in the same way before. In the search for patterns, data may lead to information; when it does, a considerable amount of data may be distilled into a much smaller (and more manageable) amount of information, producing what is most useful to the conceptual planner: real insight into the nature of a problem. This frequently can only be expressed in *soft* or qualitative terms, a form difficult to deal with by quantitative means—but most valuable for the generation of ideas.

In the **Action Analysis** process, Functions are associated with insights—about why things go wrong in performing the Functions, or about how special factors combine to allow other Functions to be performed well. These insights are documented as **Design Factors** and become part of a qualitative information file along with the Functions.

Activity Analysis forms (left in Figure 4) record information at the Activity level. Design Factor forms (right in the figure) document insightful observations and ideas associated with the Functions of an Activity.

The Activity Analysis form is divided into three sections. In the first section, at the top, the *scene* is set. Users are listed by roles or types, and system and environmental components are identified. In the



Activity Analysis — Activity: **Orienting** — 26

Project: **Access to Justice**
Mode: **Enforcement**
Originator: Holly Roeske
Contributors: 13 Feb., 2001 — Loren Gulak, Michael Heller, Jennifer Joos, Jin Lee; 4 May, 2001 — Charles Owen

Scenario: A self-represented litigant seeks to understand the enforcement process and form a strategy to pursue a judgment made in his favor.

Users *(Players)*	System Components *(Props)*	Environmental Components *(Set)*
Self-represented litigant	Forms	Lighting
Plaintiff's family/friends	Brochures	Space configuration
Court staff	Computer station	Work space
Other self-represented litigants	Printer	Storage units
Assistance program staff	Database	Computer network
Facilitators (in some states)	Work surface	Library

Functions:
155. Gather information
158. Accommodate resources
159. Build enforcement strategy
157. Weigh value of pursuit
160. Select appropriate pleading
156. Analyze information

Associated Design Factors:
136. Unable to Locate Information
138. Resources Not Consolidated
137. Space Not Provided
139. Tools Not Available
135. Unable to Assign Value to Options
140. Unable to Comprehend Material

Version 3 Date: 4 May, 2001 Date of first version: 31 January, 2001

Design Factor — Title: **Unable to Assign Values to Options** — 135

Project: **Access to Justice**
Mode: **Enforcement (Preparation)**
Activity: **Orienting**
Originator: Holly Roeske
Contributors: 16 Feb., 2001 — Loren Gulak, Jin Lee, Jennifer Joos, Michael Heller; 29 Apr., 2001 — Charles Owen

Source/s: Heller, Shapiro and Frisone, Ltd. **Notes on Collection.** Chicago: Unpublished notes, 2000.

Associated Functions: 157. Weigh value for pursuit; 160. Select appropriate pleading

Observation: When more than a few options are available to self-represented litigants for enforcing a judgment made in their favor, it can be difficult to select among the options.

Extension: Enforcement of a judgment can be a time-consuming and expensive process. Many self-represented litigants are unaware of the significant additional commitments they may have to make in order to collect on a judgment made in their favor. Assuming that a favorable judgment is the end of the process is a not-uncommon mistake of litigants working on their own behalf.

Enforcement of a judgment can be made in a number of ways including: mediation, additional court proceedings leading to the setup of payment plans, garnishment of wages, and several other means. Each procedure has different time and effort requirements. Some judgments may be so small relative to the time and resources necessary to collect them that the self-represented litigant may well consider not even pursuing collection.

Design Strategies / Solution Elements (Specify status: E Existing, M Modified, S Speculative):
Provide means of identifying options and deciding among them — S Pursuit Evaluator
Match impacts of enforcement strategies to self-represented litigant's life style — M Enforcement Checklist

Version 3 Date: 29 April, 1999 Date of first version: 15 February, 2001

Figure 4. The **Activity Analysis** *form (on the left) is used to identify* ***Functions*** *and* ***Design Factors*** *associated with an activity. It helps an analyst to cover the areas of inquiry thoroughly. A* **Design Factor** *(on the right) records insights and information about* **Functions** *(****Observation*** *and* ***Extension****) along with ideas for how to use that knowledge (****Design Strategies*** *and* ***Solution Elements****).*

sections below, Functions are listed either as actions taken by the system or actions performed with the system by users. As they are developed, Design Factors listed to the right of the Functions to which they pertain.

Formats for naming Functions and Design Factors are fixed. Since a Function is essentially an action or maintenance of a condition, the most natural way to describe it is with a verb phrase. Design Factors are about problems and insights. To make titles for them most useful, they should capture in a concise phrase the essence of the insight the analyst has realized. In that way it is most likely to remind planners accurately of the problem (or opportunity) when they see it.

The Design Factor document contains a number of entries. Its primary purpose, however, is the provision of information of two kinds: *information about the problem (or opportunity)* detected, and *information about what might be done about it*. The fact that problem and solution are both covered in the same document is not accidental. It is important that when insights are recognized, ideas be sought for how to use them. These ideas may not be used in a final concept for the system, but they are important as progenitors and are used in structuring the information file later in the process.

The ***Observation*** section is the first of two sections dealing with the problem. An Observation is a sentence in which an insight about the performance of a Function is recorded. As much as possible, it should *distill the essence* from the observed phenomenon. Frequently it is helpful to express the sentence in a condition/occurrence format. In this format, a condition is defined in a dependent clause; and an occurrence that takes place when this condition is present is described in a following independent clause. If this format is used, the conjunctions "if", "when", "while", "because", "where" or others may be helpful in introducing the condition. It is important, however, not to overstate (or overrate) the certainty of the relationship between condition and occurrence—the term *Observation* is meant to indicate that a phenomenon is observable, nothing more. A cause/effect relationship should not be inferred when, in fact, that strong a relationship cannot be justified (more than one cause may be required for the effect; the effect may be one of many and not justifiably isolated; the effect may not always follow from the cause; etc.).

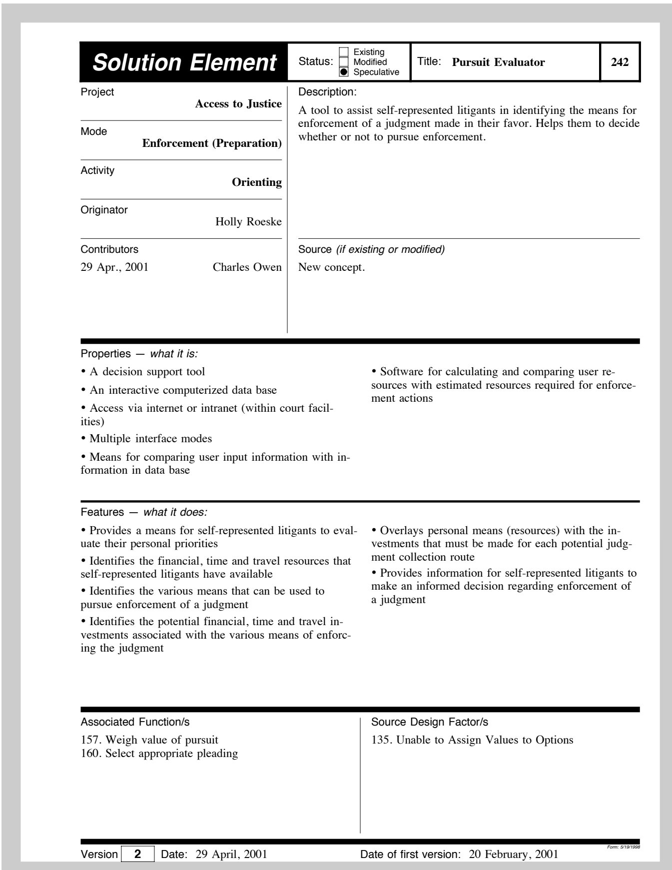
Associated with the Observation section is a section labeled ***Extension***. In this section, explanatory material is placed to extend or develop the information of the Observation. No matter how thoughtfully worded, the single sentence of the Observation seldom is enough to convey the insight adequately. The *whys* and *what-do-you-means?* that inevitably are asked are addressed in the Extension. Supplementary material from other sources may be discussed; examples may be cited; contributing phenomena other than those mentioned in the Observation may be introduced; side effects may be considered. After examining the Extension section, readers should have a good understanding of the insight of the Design Factor. They should be able to appreciate its value and, perhaps, even anticipate the directions for using it that will be suggested in the next sections.

The first of two sections dealing with ideas is the ***Design Strategies*** section. Design Strategies are, by definition, generalized suggestions for how to react to the information of the Observation and its Extension. They express the *implications* that this information has for design. For a format, they take an imperative verb phrase, carefully crafted to prescribe an approach without specifically describing a solution. Typically, Design Strategies are specialized for the situation from general strategies for problem solving such as: confront the problem, remove the cause of the problem, avoid the problem, block the problem, divert the problem, break up the problem, reduce the problem, etc.

The ***Solution Elements*** section is the second solution section. Specific ideas go into this section. Solution Elements are ideas well enough described to be evaluated as useful to the system being developed. They do not have to be original; in fact, they are distinguished as being *existing, modified* or *speculative*, depending on the level of innovation that the planning team feels that it has contributed. They are important for determining interaction among Functions (as shall be discussed) and may actually be used in the overall solution, but they should not be overly valued at the time they are written. For a name format, they take a noun phrase. Noun phrases express concepts well and are easy to remember—especially if they include colorful phraseology. A good name for a Solution Element has an adjective and a noun chosen to create an evocative title. Such a title, once explained, is readily retained in memory, and a wealth of detail associated with the concept is usually recalled with it.

Other sections on the Design Factor form serve administrative needs. The ***Originator*** section records the author of the Design Factor. The ***Associated Functions*** section ties the Design Factor to the Functions for which it was written (the title should appear as it does on the Activity Analysis forms). The ***Title*** block names the Design Factor and is the name found on the Activity Analysis form as the Associated

Design Factor for a given Function. For ***Source/s***, an entry following standard bibliography format is used (with footnote entries in the text to locate specific reference pages). If the information is derived from the Originator's direct observation or personal experience, the Source entry may read "Personal observation".



Solution Element — Status: Existing / Modified / ● Speculative — Title: **Pursuit Evaluator** — **242**

Project: **Access to Justice**

Mode: **Enforcement (Preparation)**

Activity: **Orienting**

Originator: Holly Roeske

Contributors: 29 Apr., 2001 Charles Owen

Description:
A tool to assist self-represented litigants in identifying the means for enforcement of a judgment made in their favor. Helps them to decide whether or not to pursue enforcement.

Source *(if existing or modified)*: New concept.

Properties — *what it is:*
- A decision support tool
- An interactive computerized data base
- Access via internet or intranet (within court facilities)
- Multiple interface modes
- Means for comparing user input information with information in data base
- Software for calculating and comparing user resources with estimated resources required for enforcement actions

Features — *what it does:*
- Provides a means for self-represented litigants to evaluate their personal priorities
- Identifies the financial, time and travel resources that self-represented litigants have available
- Identifies the various means that can be used to pursue enforcement of a judgment
- Identifies the potential financial, time and travel investments associated with the various means of enforcing the judgment
- Overlays personal means (resources) with the investments that must be made for each potential judgment collection route
- Provides information for self-represented litigants to make an informed decision regarding enforcement of a judgment

Associated Function/s:
157. Weigh value of pursuit
160. Select appropriate pleading

Source Design Factor/s:
135. Unable to Assign Values to Options

Version 2 Date: 29 April, 2001 Date of first version: 20 February, 2001

Figure 5. Ideas first conjectured in a **Design Factor***, are given form in a* **Solution Element.** *Intended to be brief "sketches", these capture what it is and what it does in simple notes.*

Solution Element documents (Figure 5) detail the ideas noted on Design Factors. These documents are one-page, short forms designed to capture enough detail about ideas to give them substance when they are needed later. Besides the same kinds of reference blocks used on Design Factors and Defining Statements, they have three important sections. The first, ***Description***, is for a short, one or two phrase explanation of what the Solution Element is. This is expressed at a general level and should be just enough to identify what it is and what it does at a high level. The other two sections, ***Properties*** and ***Features***, isolate the specific aspects of the idea that give it its identity.

Properties are *what it is*. Expressed in noun phrases, a series of bullet lines establish what functional enities need to be present to make the concept work. Features are *what it does*. Verb phrase bullet lines do the same thing for its benefits. Essentially, the Properties are what the design and/or engineering teams will want to know (what has to be developed), and Features are what the communications, and/or marketing teams will need (why someone will appreciate it).

The simplicity of the Solution Element form and the directness that it requires for description give it its value. In the press of analysis, observation and search for understanding, many insights unfold and many ideas emerge almost unbidden. In conventional processes, these are mostly lost for lack of any systematic way to capture them. In Structured Planning, the Solution Element form is the tool for capture.

The results of the Action Analysis process are collected in a **Function Structure** (Figure 6). The Function Structure reveals the what must be accounted for by the project in both breadth and depth, and provides a visually convenient means for judging the coverage of the analysis process. The product of the Action Analysis process is actually much more, of course. Three sets of critical information have been obtained: a set of Functions, a set of insights and a set of ideas—the latter two described in Design Factor and Solution Element documents.

Paradoxically, as useful as the Function Structure is for establishing coverage, it is not the best form of organization for developing concepts. Organizing information for use in concept development is the job of two computer programs, **RELATN** and **VTCON.** These programs incorporate specialized theory for how information should be structured for the synthesizing phase of planning.

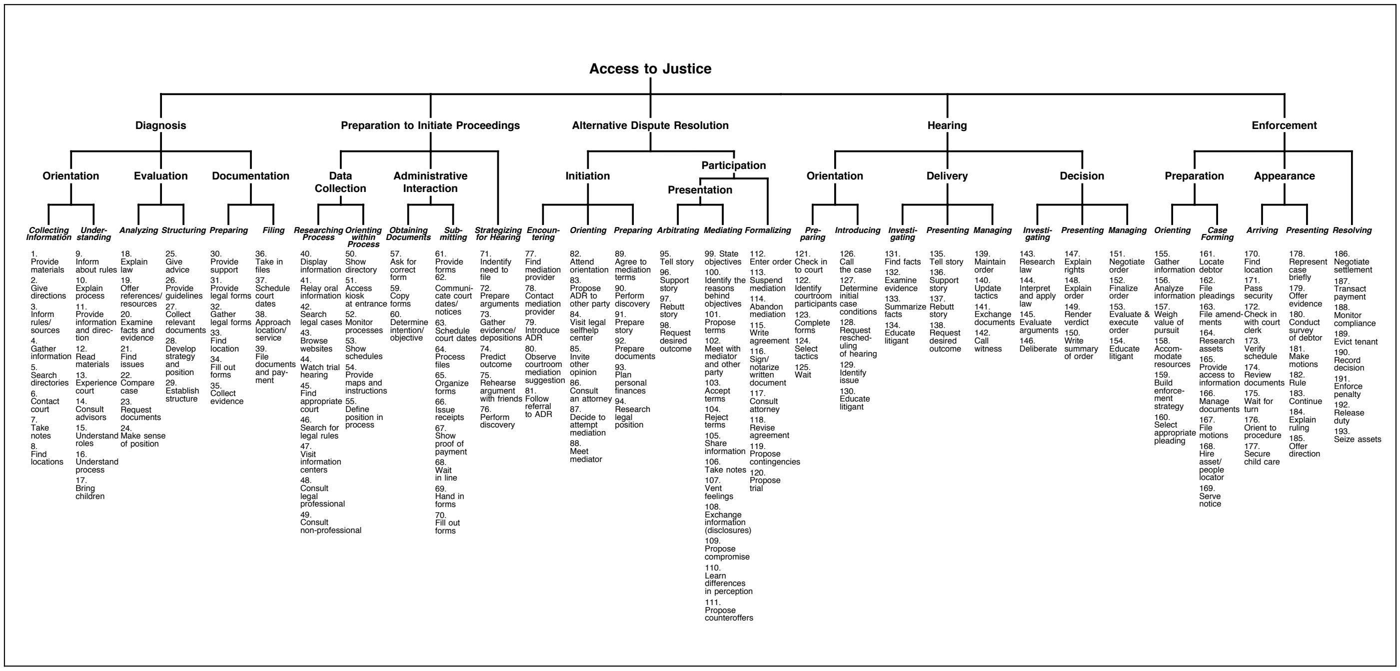


Figure 6. ***Action Analysis*** *produces a* Function Structure *by top-down examination of* **Modes** *and* **Activities.** *In this example, the Modes are in bold, the Activities are in bold italic (at the bottom of the hierarchy), and the* **Functions** *are in columns below the Activities.*

Structuring the Information I

If there are few Functions to consider, a project can be managed without much trouble. It does not take very many Functions to change that situation, however. Over 20 to 30 Functions to manage almost always means that some kind of organization must be attempted to bring order to the process. Assuming that any project of interest will have hundreds of Functions, the nature of the organizational scheme becomes a matter of importance.

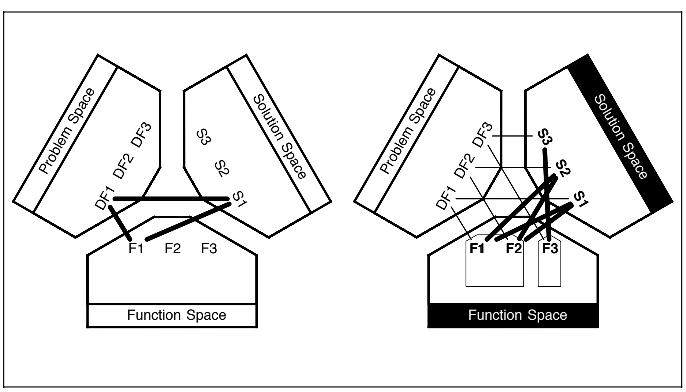


Figure 7 (left). **Functions, Solution elements** *(solution ideas) and* **Design Factors** *are related by mappings from space to space as a Design Factor records insight about a Function, a Solution Element takes advantage of a Design Factor's insight, and a Function is fulfilled by a Solution Element (or would be made difficult to fulfill by the presence of the Solution Element in the final solution). Right: the entities of one space, in this case Functions, can be clustered by their associations with entities of another.*

How should Functions be organized? The conventional way to organize almost anything is to look for similarities among the items to be classified and to put like items together. Sometimes the categories are preselected and the likenesses measured are those between items and ideal members of the categories; sometimes (as in numerical taxonomy) the categories are defined in the process by the natural grouping of like objects on a number of preselected characteristics or attributes. A number of theoretical models have been developed for the clustering of items in this way, and computer programs exist to do most of the work. The question is: is similarity, however it is employed, the best relationship to use for organizing Functions? Christopher Alexander suggested another way of thinking that leads to a much more sophisticated concept for organization.

The controlling factor for whether two Functions are related from the planning standpoint is not whether they are *alike,* but whether they share potential solutions—or, put more correctly, whether a significant number of their potential solutions are *of concern* to both Functions (Figure 7). This includes, in a sense, whether they are *unalike* because of their potential solutions. The concept, once examined, is very appealing. In the first case, if planners consider those Functions together that have a number of potential solutions *in common*—that is, a solution for one Function also, in some way, is a solution for a second Function—there is an excellent chance that they will be able to fine-tune one or a few solutions so that they will meet the requirements of the Functions under consideration very well. In the second case, if they can see Functions together that have potential conflict problems because of some of their potential solutions (a solution for one Function, if accepted for the overall system concept, aggravates or prevents meeting the needs of a second Function), they have the opportunity early-on to select or devise solutions that will avoid the difficulties.

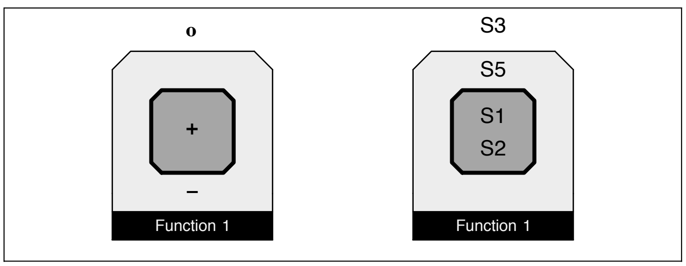


Figure 8. Located on a "bull's-eye" diagram, all **Solution Elements** *developed for a project can be categorized according to how they support (+), obstruct (-) or have no bearing (0) on fulfillment of any particular* **Function.**

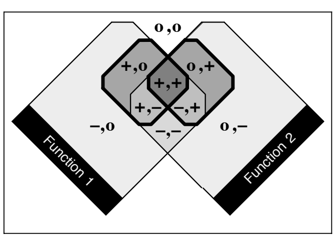


Figure 9. The combination of bull's-eye diagrams for two **Functions** *establishes all possible regions of combination of support or obstruction for both Functions. Only regions supporting one (+,-), (+,0), (-,+), (0,+) or both Functions (+,+) contain* **Solution Elements** *that might be chosen for a design.*

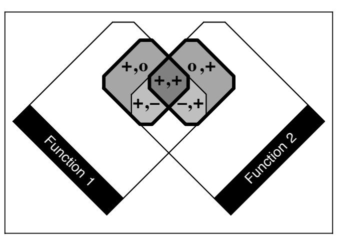


Figure 10. Intuitively, the amount of interaction between two **Functions** *is proportional to the number of* **Solution Elements** *in the three overlapping regions relative to those in all five regions from which Solution Elements might be chosen for a design.*

The **RELATN** program uses this concept to establish links between Functions based on the Solution Elements given for a project. How it does this can be illustrated with two diagrams. In the first diagram (Figure 8), the "bull's-eye" represents a two-part abstract space that contains all of the Solution Elements for a project that in some way are of concern to a Function (Function 1, for example). The diagram has a bull and a ring because some of the Solution Elements help to *fulfill* Function 1 (+), and some—if they are used to fulfill other Functions in the project—will make it *difficult to fulfill* Function 1 (-). Both kinds of Solution Elements are obviously of concern. There are, of course, other Solution Elements in the collection for the whole project; they are represented in this diagram as being outside the bull's-eye space (0), because they have no bearing on Function 1—they neither support nor obstruct its fulfillment. On the left in Figure 8, the spaces are shown; on the right, the Solution Elements of Figure 7 have been inserted for Function 1.

In the diagram of Figure 9, a similar bull's-eye for Function 2 is combined with that for Function 1. The intersection of the two creates regions with all the possible combinations of the characteristics from the two original bull's-eye diagrams. The pairings of positive, negative and zero values indicate the support or obstruction the Solution Elements in each region exhibit for the Functions: left position for Function 1, right for Function 2. The five regions of importance are those which contain the *positive* Solution Elements, in other words, all the solutions that might be selected to fulfill either of the two

Functions. Using these five regions, the amount of *interaction* between the two Functions (the degree to which the two Functions are related) can be established.

In the (+,+) region are the Solution Elements that fulfill both Functions. These are, in a way, the elegant solutions because each fulfills both Functions at once. The (+,0) and (0,+) regions also contain Solution Elements that might be used with confidence. Two Solution Elements, one from each of these regions, would create a total solution for the two-Function system. While not as elegant, this set of choices at least does not introduce difficulties and, in fact, the independence thus identified may be important in some planning considerations. The two remaining regions, (+,-) and (-,+), are troublesome. A Solution Element chosen from either will create a situation in which it will be difficult to successfully fulfill the Function for which the (-) value was given. Based on the effect they have on the two Functions, the five regions are labeled: reinforcement (+,+); independence (+,0) and (0,+); and conflict (+,-) and (-,+).

The concept of interaction can be drawn intuitively from the diagram. Assuming that the reason two Functions should interact (or be linked) is that they have potential solutions of concern in common, the amount of interaction should be proportional to the number of Solution Elements in the *common* regions of reinforcement and conflict relative to those in all five regions including those and the two independence regions (Figure 10). None of the other regions is relevant because no Solution Element would be chosen from them to fulfill either Function. Thus, in its simplest form, a measure for interaction is the ratio of the number of reinforcing and conflicting Solution Elements to those plus the number of independent Solution Elements.

In the **RELATN** program, the interaction concept is extended with three additions. First, instead of simply counting the presence of Solution Elements in a region, the program accepts scaled evaluations for how much a Solution Element supports or obstructs fulfillment of a Function. Scales may be of any resolution, but usually have five values: strongly supports (+2), supports (+1), no bearing (0), obstructs (-1) and strongly obstructs (-2).

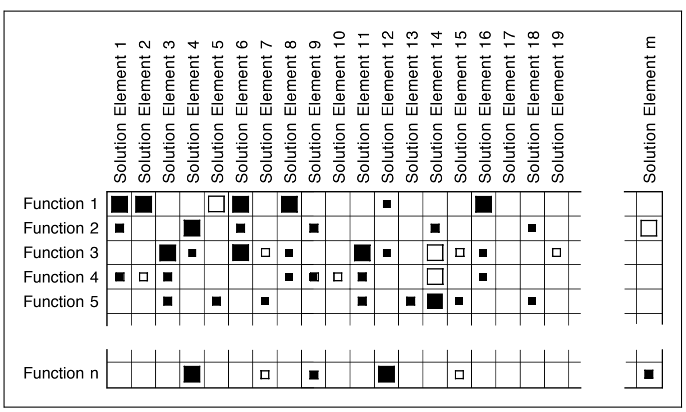


Figure 11. **Solution Elements** *are assessed for their potential support (solid marks) or obstruction (hollow marks) of* **Functions** *to establish the data for determining interaction among Functional pairings.*

Second, weights are accepted for the Solution Elements. With weighting, the impact of any Solution Element can be increased or decreased in its effect on the amount of interaction. Weights typically are used to reflect the likelihood that a Solution Element will be used in the final system solution—some ideas are more practical than others, for example; or some may be favored or even required by constraints placed on the project.

Finally, a balancing factor is incorporated to take care of the problem that some Functions have more Solution Elements of concern than others. The problem arises when a Function with only one or two positive Solution Elements is considered with one that has many (fifty would not be uncommon). If they have one common Solution Element in the reinforcement or conflict regions, what should the amount of interaction be? Intuitively, it is different depending on which Function's viewpoint is chosen. The balancing factor finds a middle ground.

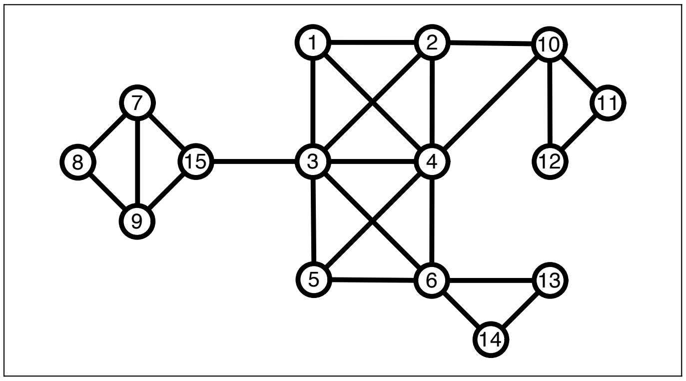


Figure 12. Links determined by the **RELATN** *program indicate which* **Functions** *(vertices) have high enough levels of interaction to suggest consideration together (the graph in this example has been optimally arranged for visual inspection).*

To prepare for using the **RELATN** program, the planning team assesses the collective set of Solution Elements against the set of Functions (Figure 11). Data for each Solution Element includes its name, weight and the scale used to assess it (different scales can be used for each Solution Element—although, in practice, a common scale is usually used for all). Data for each Function includes the Function's name and value assessments for how all the Solution Elements support or obstruct it. Experience has shown that the considerable job of assessment can be made manageable by splitting up the task among the team members. The Functions are divided up among two-member subteams. Each subteam assesses all Solution Elements for its subset of the Functions. Both subteam members independently do the entire assessment for their subteam's Functions and then compare results. Consultation (the greatest time demand) is, therefore, only required for disagreements. The loss of accuracy (agreement of the results with what would have been derived from a full-team consensus on each assessment) has been acceptably small in test comparisons.

The result of operations with the **RELATN** program is a nondirected graph, or network, in which Functions are the vertices (or nodes). Links between Functions indicate which Functions have enough interaction to warrant being considered together in any conceptual development activity (Figure 12). For many purposes, this level of organization is sufficient; but for most planning projects, further structuring is valuable.

Structuring the Information II

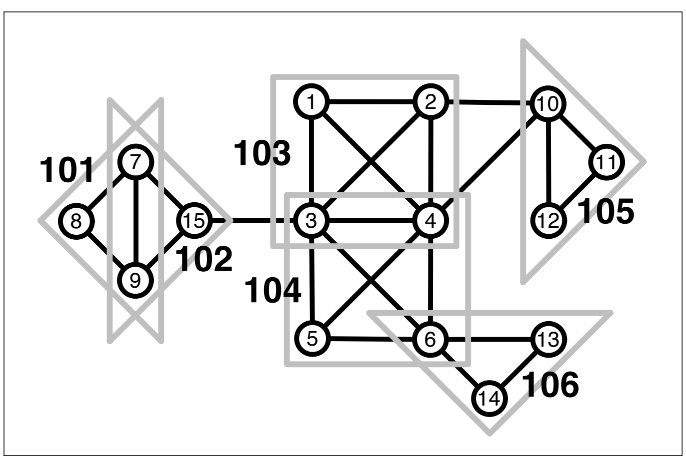


Figure 13. As its primary task, the **VTCON** *program locates clusters of vertices that are heavily interlinked (***Functions** *in the Structured Planning application). Cluster notation denotes both level and cluster number (i.e.,* **103** *is level* **1** *cluster* **3***).*

Another program, **VTCON,** is called into play to provide additional structure beyond that inherent in the graph. The graph establishes paths through the Functions by linking Functions when they are related to each other, but, unlike a road map, a graph is not necessarily arranged nicely for visual inspection. As it is obtained from the **RELATN** program, a graph is only a list of what Functions are linked to what other Functions. To draw out the analogy, it is like being in a town and having a list of towns that are next on each road out of town, but not being able to find out whether any of those towns have roads between them without going to one of them or consulting a similar list of roads for each town. If a bird's eye view were possible, clusters of towns interconnected by roads would be obvious. Unfortunately, for complex graphs, endless visual interpretations are possible, and it is extremely difficult to show one as an optimally arranged "*map*". What can be done—and what the **VTCON** program does—is to find the clusters of Functions (vertices) algorithmically (Figure 13). With that information, the purposes of the map can be achieved.

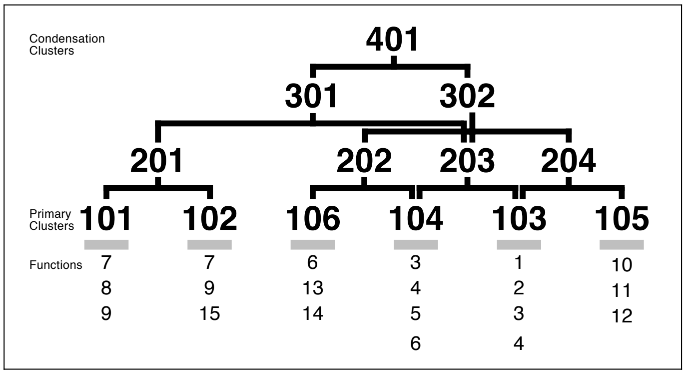


Figure 14. Continued clustering at successively higher levels by the **VTCON** *program produces a hierarchical organization of the entire set of* **Functions—an** Information Structure *useful for concept development and evaluation.*

The clusters are important because they represent primary groupings of Functions. Once the clusters have been found, the planner can choose a Function at will and know which other Functions are of direct concern. Of course, Functions are also linked to others outside their primary clusters or the graph would be unnaturally disjoint. These cross-cluster links provide the basis for higher level, broader-reaching clustering, and **VTCON** uses them to create a *condensation* hierarchy (Figure 14). Clusters are themselves clustered based on Functions held in common and links between Functions in different clusters. Levels of hierarchy are produced with smaller numbers of larger clusters at each succeeding level until the entire graph is condensed into a final cluster, the original set of all Functions. In form, the hierarchical structure is a semi-lattice rather than a tree because Functions can be in more than one cluster and clusters can be themselves members of more than one

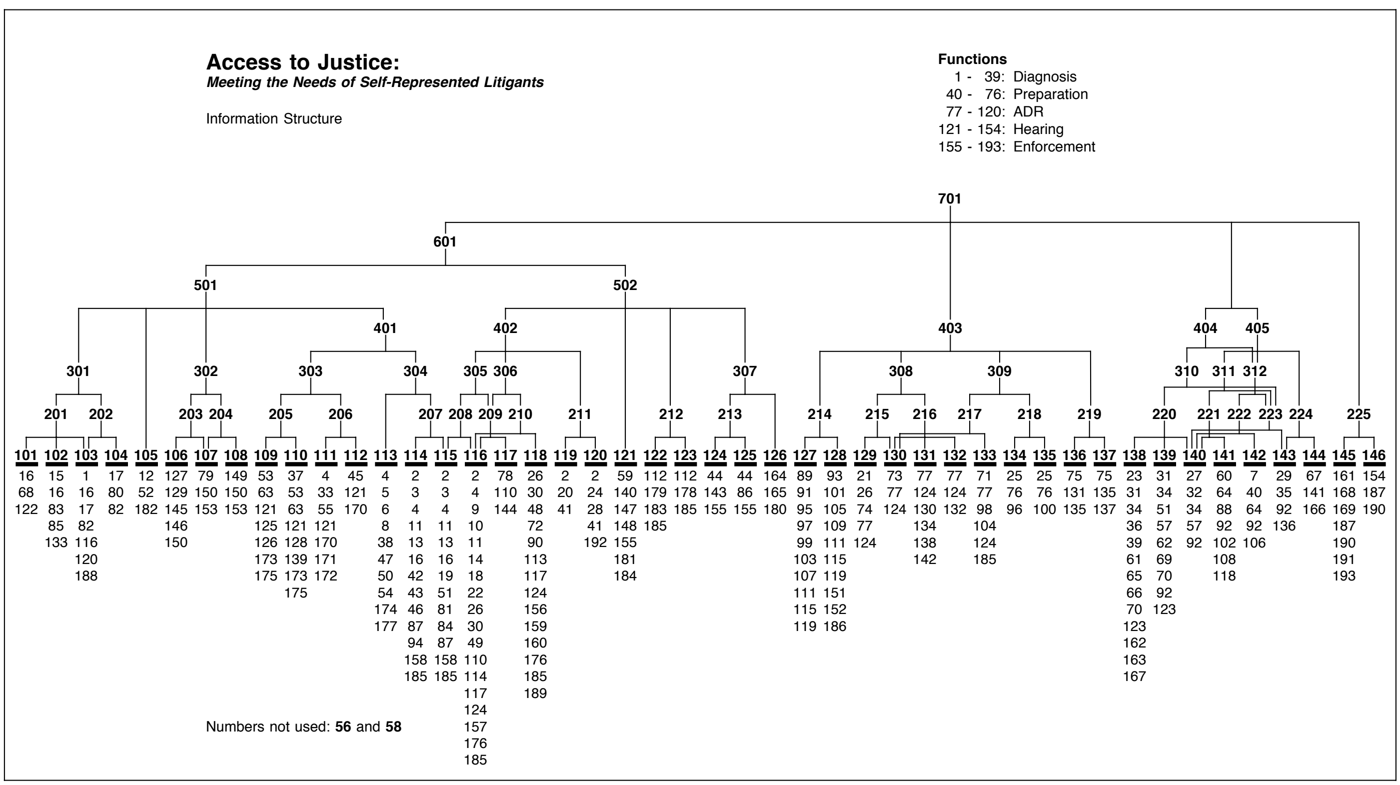


Figure 15. The **Information Structure** *produced by the* **VTCON** *program for the* ***Access to Justice*** *project.*

higher level cluster. This is a very general form of hierarchy and one most appropriate for planning—where it is natural to expect a Function to be performed in more than one Activity. Functionally, the hierarchy is an **Information Structure,** a specialized structure for synthesis. The actual Information Structure developed for this project is shown in Figure 15.

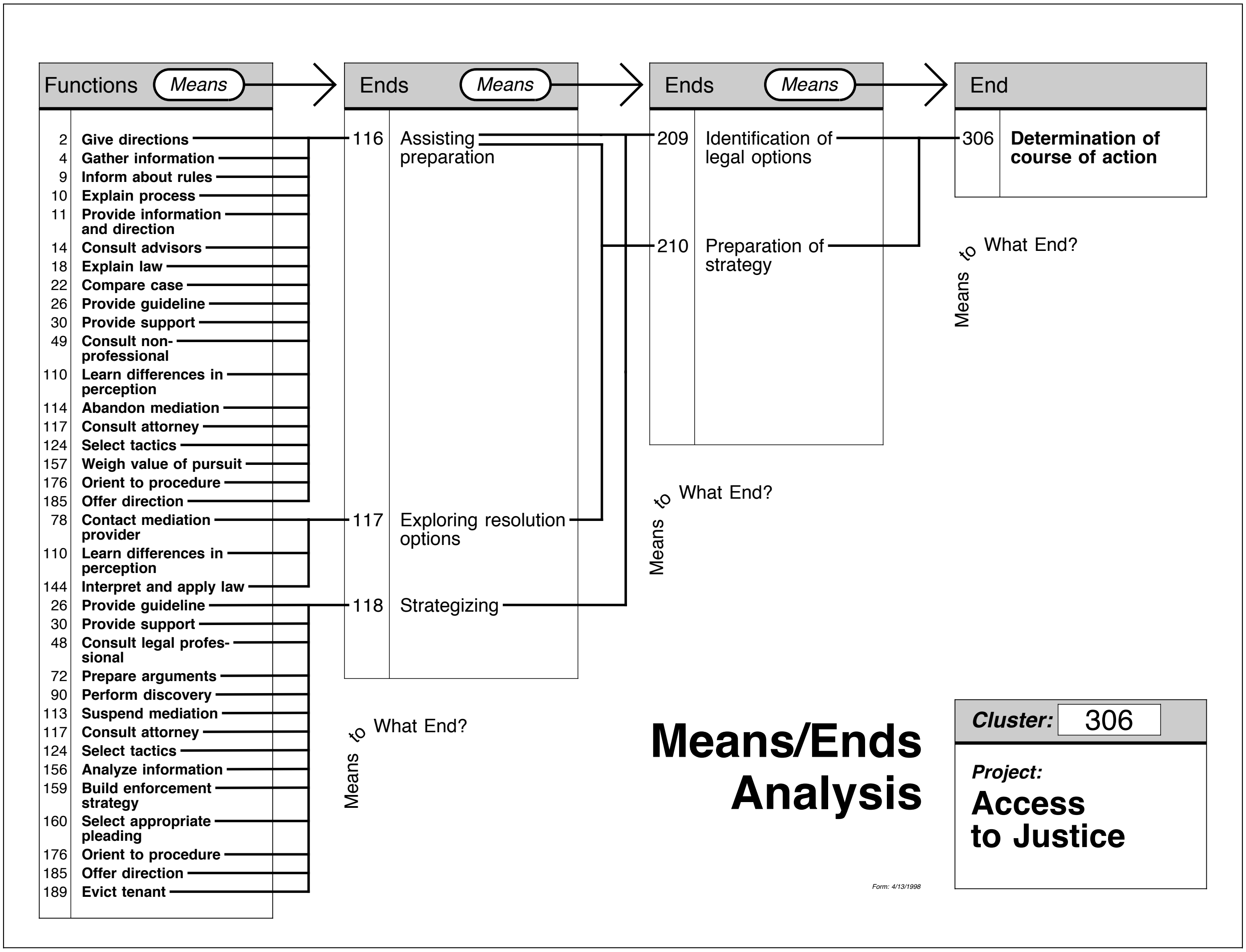


Figure 16. A cluster from the Information Structure *(306) is subjected to Means/Ends Analysis to establish meaning for the structure. Beginning with the* **Functions** *at the left, clusters are given labels that express the functionality of the structure as insightfully as possible.*

Using the Information

The results of the **VTCON** program are given in three parts: (1) a list of the primary clusters with their component Functions, (2) a compilation of links within these clusters and links between clusters as they are revealed in condensing clusters at succeedingly higher levels of the hierarchy, and (3) the Information Structure, a listing of the hierarchy giving the clusters at each level by code name (e.g, 302, meaning "level 3, cluster 2") with their next-lower-level component clusters. This information enables the Functions, Design Factors and Solution Elements to be brought together for optimal support of the ensuing processes of synthesis.

Several means for synthesis have been developed in Structured Planning. Each has certain strengths, and combinations are possible.

The technique used for this project reconstructs a traditional idea-generating process, Means/Ends Analysis, as two complementary processes: **Means/Ends Analysis** and **Ends/Means Synthesis**. To begin, a cluster of workable size is selected from the Information Structure and transferred as structure (subcluster numbers and membership information) and Functions (list) to a Means/Ends form (Figure 16).

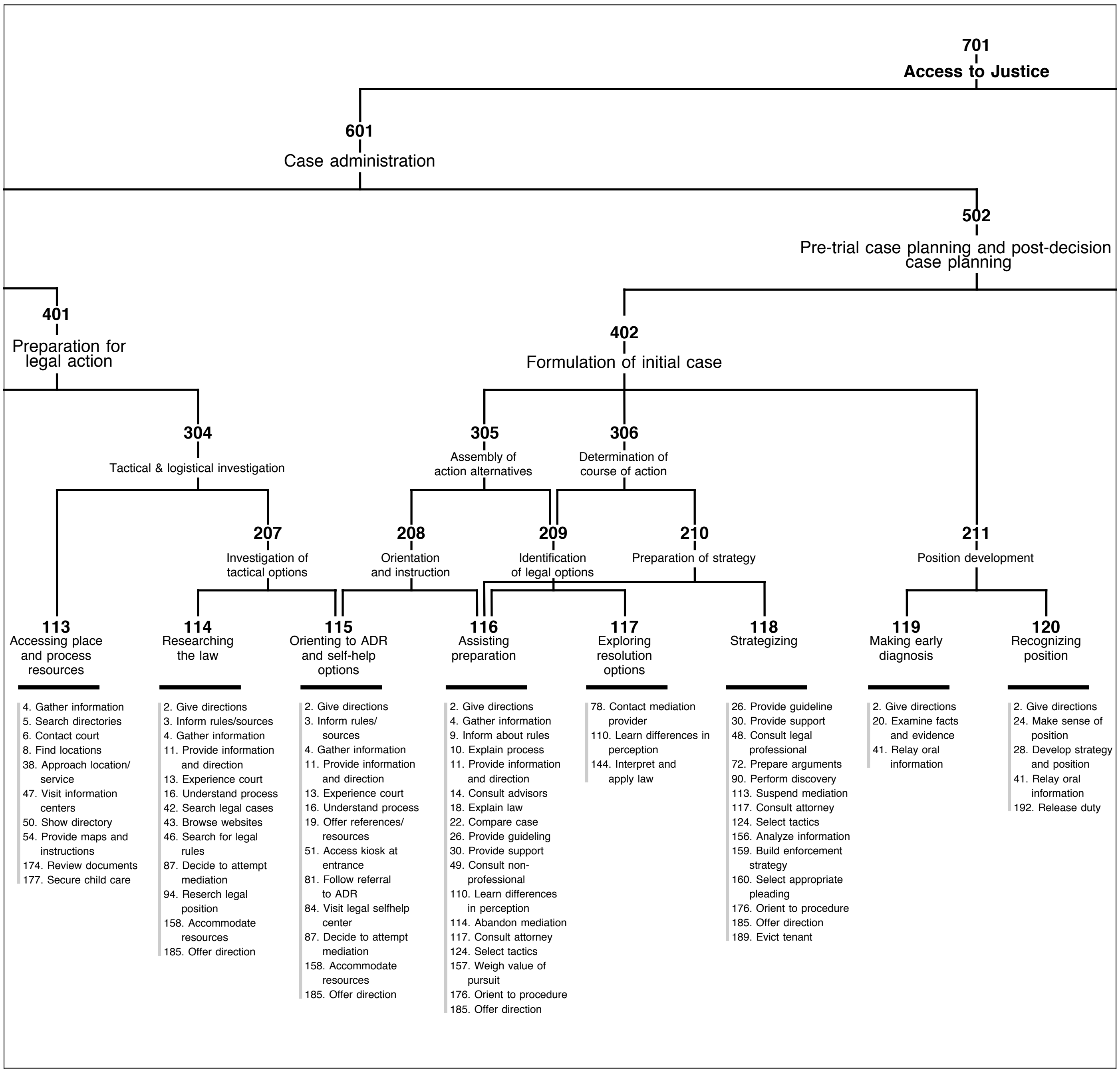


Figure 17. A segment of the named **Information Structure** *show the results of the* **Means/Ends Analysis** *process. Function names filled in, along with named clusters, help the team to see order and pattern.*

The task of Means/Ends Analysis is to create labels for all clusters. Moving from left to right through the subclusters, the question is asked, "To what *end* are these Functions *means*?" The answer is *purpose* expressed in the format for an Activity or, at higher levels, a Mode or Submode of operation.

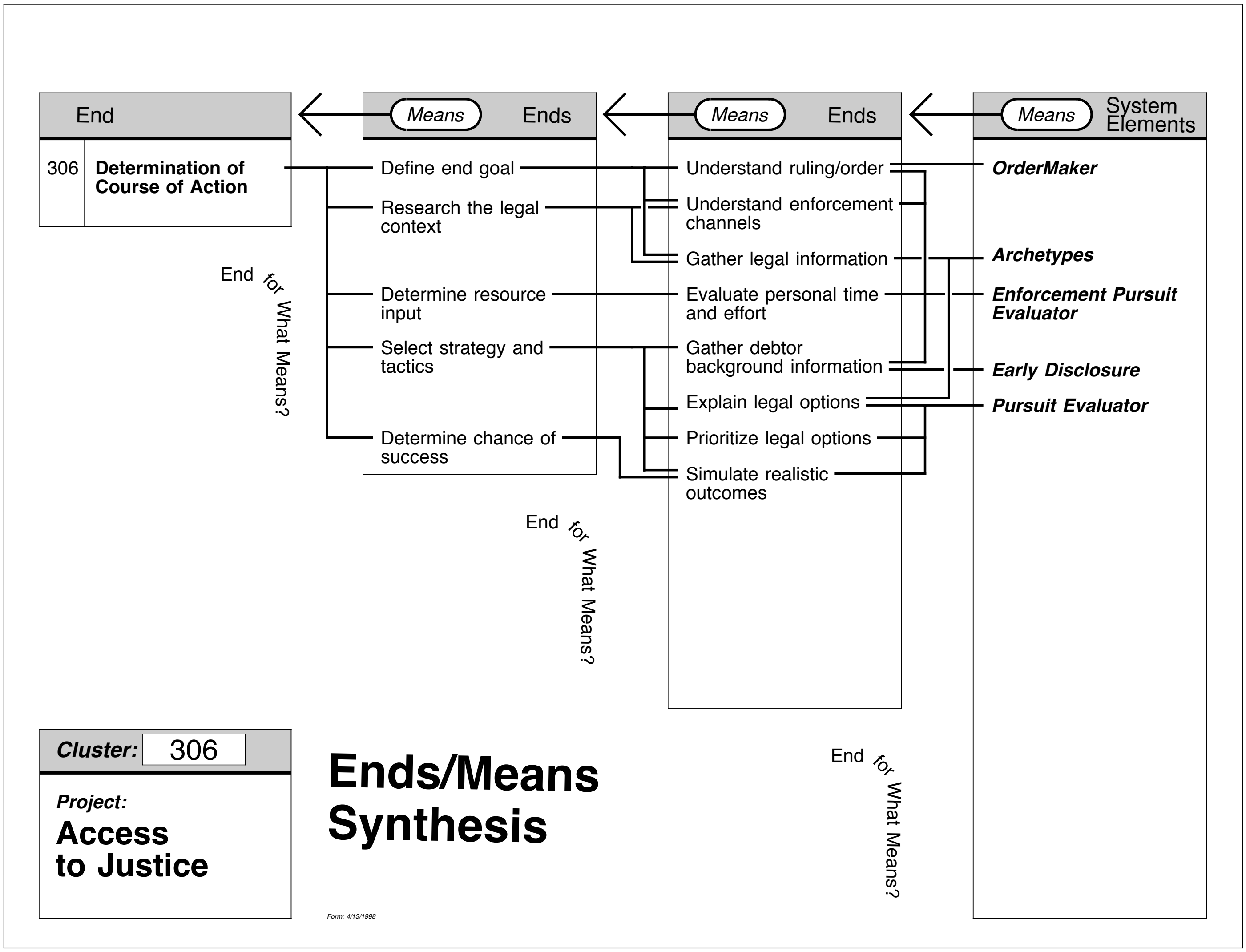


Figure 18. Using **Ends/Means synthesis***, the nodes of the Information Structure are subjected to structured brainstorming. Key nodes are established as ends, and means are sought for them with progressively sharpening focus until specific* **System Element** *concepts are identified that satisfy the needs. System Elements may be newly invented or selected from existing Solution Elements that have been refined or modified.*

When the wording of all the labels has been fine-tuned in the context of a completely labeled Information Structure (see a partial example in Figure 17), clusters are subjected to Ends/Means Synthesis. In this process, just the opposite activity occurs. Where the essence of the Means/Ends Analysis was the "discovery" of purpose seen freshly, the essence of the Ends/Means Synthesis is the "invention" of concepts to accomplish these purposes. In Figure 18, the same cluster given labels in Figure 16 is now re-examined as a challenge for invention. The highest level "purpose" is treated as the ultimate end to be reached, and the question is asked, "What *means* would meet this *end*?" New means are then generated left to right, increasing in specificity as preceding means are treated as new ends. Much as Design Strategies are treated in Design Factor documents, means are best stated as imperative verb-phrase "strategies". When ideas for means become specific enough to be final Elements of the solution package, they are given evocative noun-phrase titles (as Solution Elements were) and status as **System Elements.**

Labels given for subclusters at intermediate levels in the Means/Ends Analysis of the chosen cluster are checked for coverage as the Ends/Means Synthesis progresses, and Solution Elements originally conceived for the Functions involved are constantly reviewed as possible end products. New ideas, however, are encouraged, and original ideas may be modified or combined in the light of the ends/means that evolve.

Project: Access to Justice **Cluster:** 306 **Page:** 1

System Elements	Enforcement Pursuit Evaluator								Order Maker				Pursuit Evaluator				Archetypes						Early Disclosure					
Features / Functions	1	2	3	4	5	6	7	8	1	2	3	4	1	2	3	4	1	2	3	4	5	6	1	2	3	4	5	6
78 Contact mediation provider																▪					▪							
110 Learn differences in perception				▪	▪				▪	▪				▪	■						▪			▪			▪	
144 Interpret and apply law	▪				▪	■			▪		▪	▪		▪				▪			■							
26 Provide guidelines	■	▪	▪	■	▪					▪		▪	▪	▪		▪	▪	■			■			■				
30 Provide support	▪	▪	▪	■	▪	▪	▪		■	▪	▪	■	■	▪	■	■	▪	▪	■	▪	▪	■		■	▪	■		■
48 Consult legal professional												▪									▪							
72 Prepare arguments	▪	▪						▪				▪					▪	▪										
90 Perform discovery				▪	▪						▪												■	■		■		■
113 Suspend mediation																▪					■							
117 Consult attorney																▪					▪							
124 Select tactics	▪	■		■	▪	■	■	■					■	■	▪		▪	▪			■		▪					
156 Analyze information	▪	■	▪	▪			■	■			▪	▪	▪	▪	▪	▪		▪					▪	▪		▪	▪	
159 Build enforcement strategy	▪	▪		■	▪	▪	■	■					▪	■									▪				▪	▪
160 Select appropriate pleading		▪				▪	■		▪				▪	▪		▪		▪										
176 Orient to procedure	▪		■	■	■		▪		▪	■	▪		■	▪			■	■			■							
185 Offer direction	■	■	▪	■	■	▪	■		▪			▪	▪	▪	■	▪	■	■			■		▪				▪	
189 Evict tenant							▪		▪																			

■ Strongly supports fulfillment of the Function ▪ Supports fulfillment of the Function

Form: 4/13/1998

Figure 19. Features of the **System Elements** *are cross-checked against* **Functions** *for each key cluster to how needs are being met. Large squares indicate strong contribution to fulfillment, small squares indicate partial contribution.*

What remains is to describe the properties and features of the System Elements, ensure that there are ideas to fulfill all the Functions, and consider the System Elements against each other to draw out all systemic properties that can be gained. For the first of these tasks, the team begins to fill out what will become a System Element form (Figure 21). Although this task will have to be addressed later for completion, it is usually best to collect properties and features for an idea at the time the idea develops. Elaborations can be made at any time—if something has been recorded to elaborate upon.

The second task, checking features against required Functions, is accomplished on a tabular form, shown in Figure 19. Features are evaluated here for their contribution to fulfilling the Functions present in the primary clusters of that part of the Information Structure being addressed in the Ends/Means process. If a feature contributes significantly to fulfilling a Function, the feature/Function cell is marked boldly; if there is some contribution, the cell is marked, but less boldly. In practice, a three-option decision scheme (significant contribution, some, none) works well. A special value of this activity is that, in the process of considering how a feature of an idea may help to fulfill a Function, the thought process about how that specific fulfillment occurs often helps to crystallize the nature of the feature and the properties that generate it. Additional features may also occur to the team at this time and, of course, if there are Functions for which there are no System Elements, this is the signal to return to the Ends/Means process for more work.

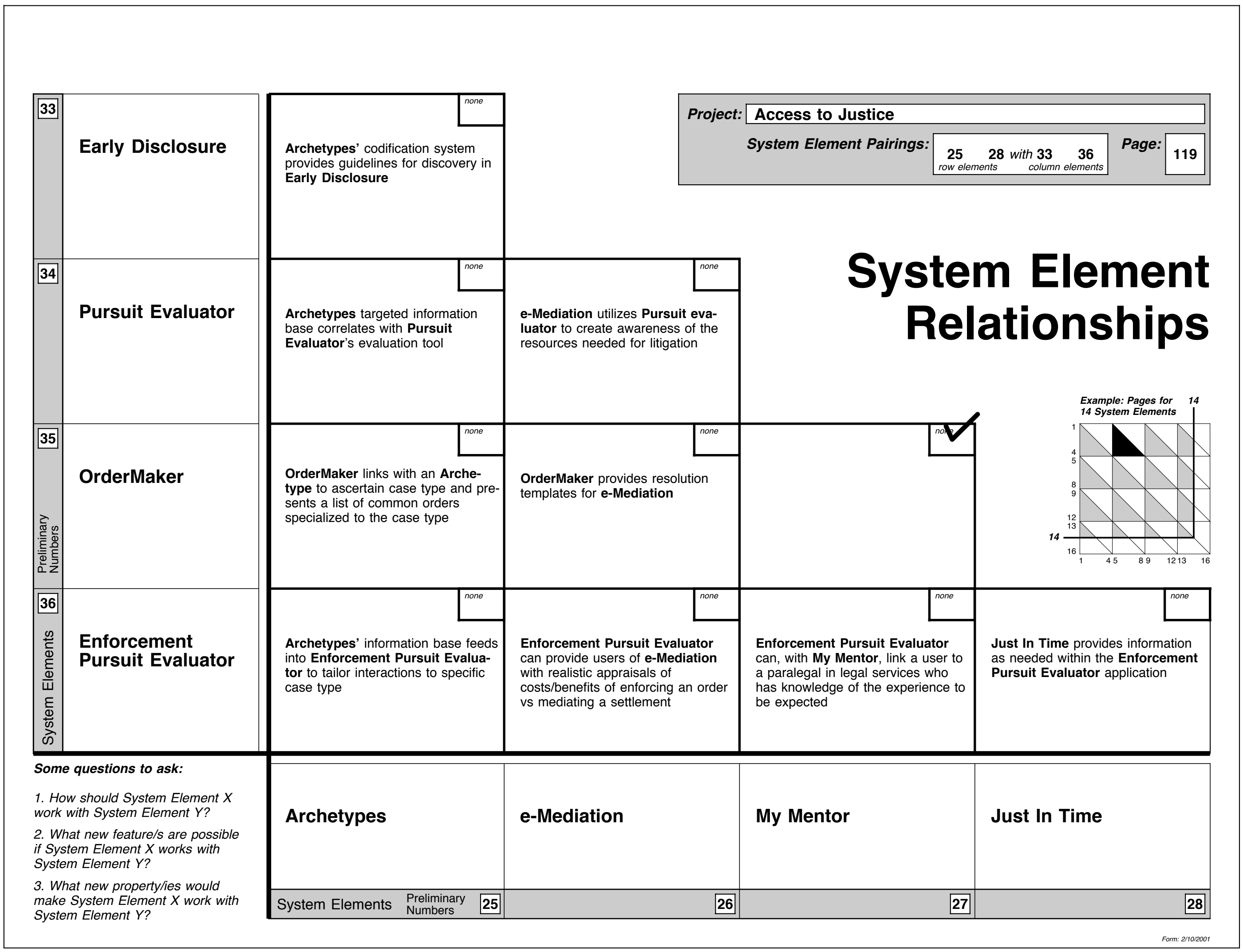


Figure 20. Systemic associations are strengthened and created using **System Element Relationships** *worksheets. The direct confrontation of System Elements with each other generates ideas for how they can work together more fully. All pairings can be examined systematically, or groupings (as in this case) can be explored for special associations.*

Finally, the third task pits System Element against System Element in a search for additional synergies that can contribute to systemic qualities. At this stage, although the Ends/Means process is complete, it is still possible to mold System Element properties and features in ways to optimize system functionality. Figure 20 shows a form used to consider System Elements four at a time against four others. The boxes in the form are used to note ways in which the pair of System Elements can work together. Rather than simply recognizing relationships, the planning team proactively seeks out ways for the System Elements to work together—to the extent of modifying one or the other, or both, to create synergy. Any changes are incorporated in the properties and/or features of the individual System Elements. At this stage of the synthesis process, when the system is at a high level of description and the team knows more about it than it ever has, it is the best possible time to extend ideas to higher levels of cooperation. The systematic consideration of relationships is a powerful creative tool.

The organization provided by the Information Structure and the synthesis support processes for using it give the planning team the bird's eye views they need of the problem. Information is juxtaposed *insightfully* with effectiveness well beyond the capability of conventional information retrieval systems. The effect is having at hand not only what you *need to know,* but also what you *didn't know you needed to know!*

Communicating the Concept

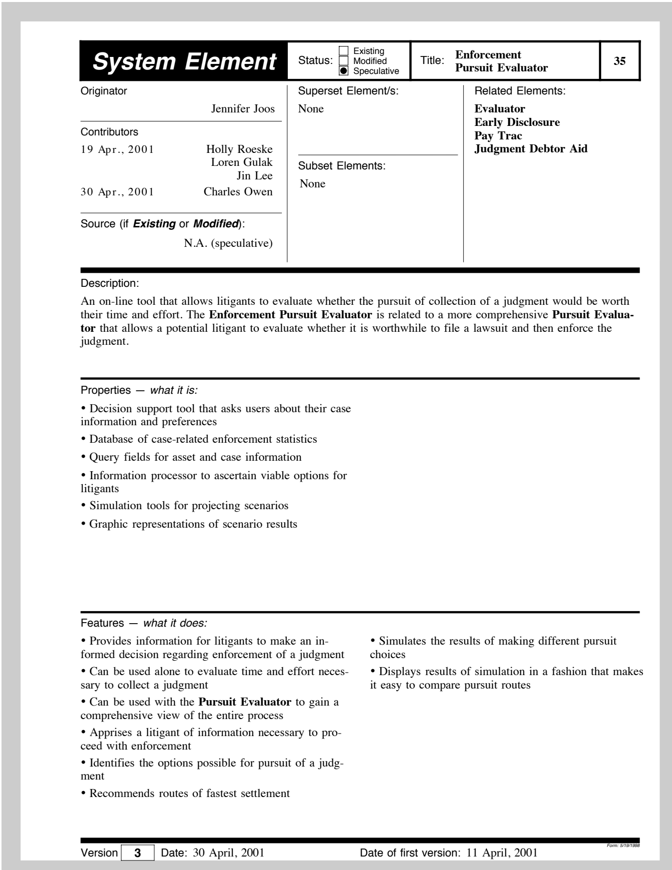
The product of the Structured Planning process is a **Plan**, made up of **System Elements** (Figures 21, 22 and 23) that describe the ideas developed to meet the needs of the project as they are outlined in the Charter and Defining Statements and refined through the Action Analysis process. Each System Element has five major parts:

Title. The title is no more than a few words (two or three, typically), in a noun phrase that captures the essence of the System Element. A good title is unique and memorable.

Related System Elements. Other System Elements that ought to be read with this one are listed in this section. The best grasp of a complex concept is achieved when ideas are appreciated in a meaningful order. Especially when there are large numbers of System Elements, there is a need to know which are strongly associated. Establishing the multiple relatedness of Elements is a hypertext concept; it allows the Plan to be examined in more than one way—with options suggested, but the actual order determined by the reader. For a large number of System Elements, the structure of association can be further extended by using **VTCON** to create a hierarchical *Communication Structure* in which clusters and hierarchy are established under the relation, "should be considered together".

Superset Elements and Subset Elements. In the process of organizing the System Elements (possibly using VTCON), it is frequently possible to group them hierarchically. The System Element form has provisions for indicating higher and lower level associations where they exist as superset or subset relationships.

Properties. Expressed in the same noun-phrase, bullet format as they were for Solution Elements, Properties are *what it is*. Together with Features, these are the essential "specifications" for what the System Element must be and do.



System Element | Status: ☐ Existing ☐ Modified ● Speculative | Title: **Enforcement Pursuit Evaluator** | **35**

Originator
Jennifer Joos

Contributors
19 Apr., 2001 Holly Roeske, Loren Gulak, Jin Lee
30 Apr., 2001 Charles Owen

Source (if ***Existing*** or ***Modified***):
N.A. (speculative)

Superset Element/s:
None

Subset Elements:
None

Related Elements:
Evaluator
Early Disclosure
Pay Trac
Judgment Debtor Aid

Description:

An on-line tool that allows litigants to evaluate whether the pursuit of collection of a judgment would be worth their time and effort. The **Enforcement Pursuit Evaluator** is related to a more comprehensive **Pursuit Evaluator** that allows a potential litigant to evaluate whether it is worthwhile to file a lawsuit and then enforce the judgment.

Properties — *what it is:*

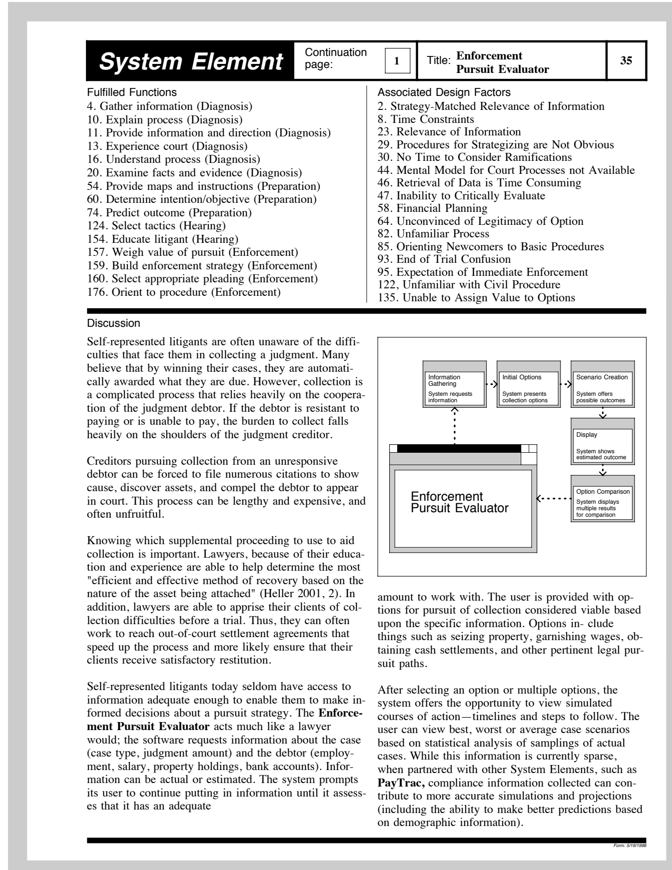
- Decision support tool that asks users about their case information and preferences
- Database of case-related enforcement statistics
- Query fields for asset and case information
- Information processor to ascertain viable options for litigants
- Simulation tools for projecting scenarios
- Graphic representations of scenario results

Features — *what it does:*

- Provides information for litigants to make an informed decision regarding enforcement of a judgment
- Can be used alone to evaluate time and effort necessary to collect a judgment
- Can be used with the **Pursuit Evaluator** to gain a comprehensive view of the entire process
- Apprises a litigant of information necessary to proceed with enforcement
- Identifies the options possible for pursuit of a judgment
- Recommends routes of fastest settlement
- Simulates the results of making different pursuit choices
- Displays results of simulation in a fashion that makes it easy to compare pursuit routes

Version 3 Date: 30 April, 2001 Date of first version: 11 April, 2001 Form: 5/19/1998

Figure 21. As elements of a Plan, **System Elements** *present individual concepts describing ideas and specifying essential properties and features.*



System Element | Continuation page: 1 | Title: **Enforcement Pursuit Evaluator** | **35**

Fulfilled Functions
4. Gather information (Diagnosis)
10. Explain process (Diagnosis)
11. Provide information and direction (Diagnosis)
13. Experience court (Diagnosis)
16. Understand process (Diagnosis)
20. Examine facts and evidence (Diagnosis)
54. Provide maps and instructions (Preparation)
60. Determine intention/objective (Preparation)
74. Predict outcome (Preparation)
124. Select tactics (Hearing)
154. Educate litigant (Hearing)
157. Weigh value of pursuit (Enforcement)
159. Build enforcement strategy (Enforcement)
160. Select appropriate pleading (Enforcement)
176. Orient to procedure (Enforcement)

Associated Design Factors
2. Strategy-Matched Relevance of Information
8. Time Constraints
23. Relevance of Information
29. Procedures for Strategizing are Not Obvious
30. No Time to Consider Ramifications
44. Mental Model for Court Processes not Available
46. Retrieval of Data is Time Consuming
47. Inability to Critically Evaluate
58. Financial Planning
64. Unconvinced of Legitimacy of Option
82. Unfamiliar Process
85. Orienting Newcomers to Basic Procedures
93. End of Trial Confusion
95. Expectation of Immediate Enforcement
122, Unfamiliar with Civil Procedure
135. Unable to Assign Value to Options

Discussion

Self-represented litigants are often unaware of the difficulties that face them in collecting a judgment. Many believe that by winning their cases, they are automatically awarded what they are due. However, collection is a complicated process that relies heavily on the cooperation of the judgment debtor. If the debtor is resistant to paying or is unable to pay, the burden to collect falls heavily on the shoulders of the judgment creditor.

Creditors pursuing collection from an unresponsive debtor can be forced to file numerous citations to show cause, discover assets, and compel the debtor to appear in court. This process can be lengthy and expensive, and often unfruitful.

Knowing which supplemental proceeding to use to aid collection is important. Lawyers, because of their education and experience are able to help determine the most "efficient and effective method of recovery based on the nature of the asset being attached" (Heller 2001, 2). In addition, lawyers are able to apprise their clients of collection difficulties before a trial. Thus, they can often work to reach out-of-court settlement agreements that speed up the process and more likely ensure that their clients receive satisfactory restitution.

Self-represented litigants today seldom have access to information adequate enough to enable them to make informed decisions about a pursuit strategy. The **Enforcement Pursuit Evaluator** acts much like a lawyer would; the software requests information about the case (case type, judgment amount) and the debtor (employment, salary, property holdings, bank accounts). Information can be actual or estimated. The system prompts its user to continue putting in information until it assesses that it has an adequate

amount to work with. The user is provided with options for pursuit of collection considered viable based upon the specific information. Options in- clude things such as seizing property, garnishing wages, obtaining cash settlements, and other pertinent legal pursuit paths.

After selecting an option or multiple options, the system offers the opportunity to view simulated courses of action—timelines and steps to follow. The user can view best, worst or average case scenarios based on statistical analysis of samplings of actual cases. While this information is currently sparse, when partnered with other System Elements, such as **PayTrac,** compliance information collected can contribute to more accurate simulations and projections (including the ability to make better predictions based on demographic information).

Form: 5/19/1998

Figure 22. Page 2 of the **System Element** *contains track-back information to* **Functions** *fulfilled and formative* **Design Factors**. *Discussion of ideas helps to fill out understanding for design teams to follow.*

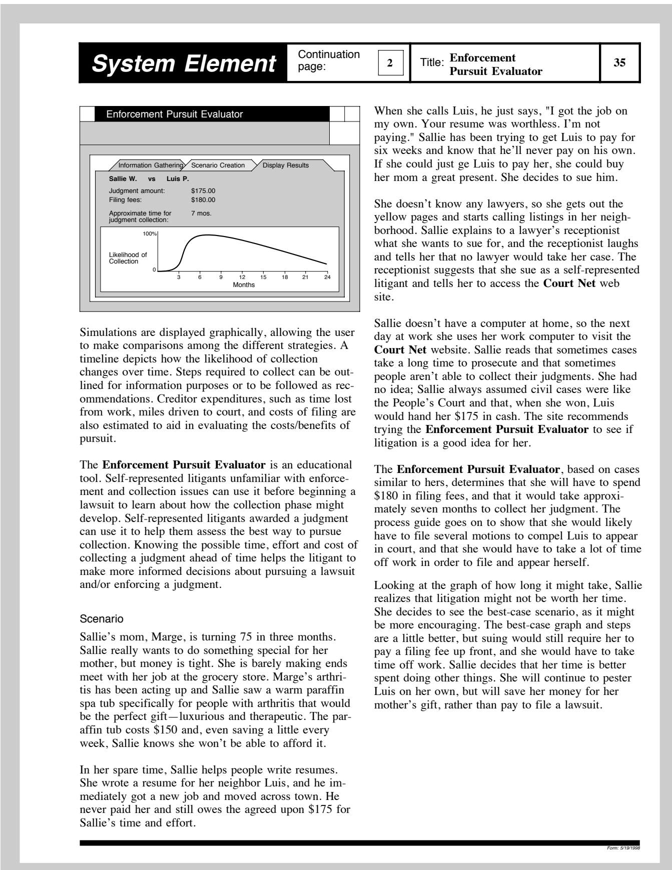
Features. These are verb-phrase, bullet lines highlighting the special functions that the System Element performs—*what it does*. They point out what is expected of the final product in as general terms as possible; specifying without over-specifying. A balancing act is required here (as well as for Properties) to provide sure guidelines without taking away too much of the maneuvering room required for creative work by the follow-on design team charged to develop the details.

Building on the hill-climbing metaphor often used in optimization theory, good Properties and Features will keep the design team climbing the right hill, but will let them find their own best path to the top.

Fulfilled Functions. This section simply lists the Functions (from the entire Function list) that the System Element fulfills. The Function list allows the design team to track the solution back to the Functions that were considered by the concept development team.

Associated Design Factors. Along with Fulfilled Functions, this section provides "track-back" information that helps the design team to understand the motivating insights that led to the ideas incorporated in the System Element.

Discussion. A full narrative description of the idea is given in the discussion section, including reasons for why the form evolved as it did. The concept development team uses this section to provide all the detail that has surfaced in the planning process, even though the purpose of the Plan is to express concept rather than detail. In effect, what is said to the design teams who will continue on is: "*Use this if you don't come up with better ideas*". Diagrams, mathematical analyses, drawings, photographs—even video clips and animations, if the medium of the Plan can support them—may be used here to supplement text. The goal is to make the description as helpful as possible. No limit exists for the discussion section.


System Element | Continuation page: 2 | Title: **Enforcement Pursuit Evaluator** | 35

Enforcement Pursuit Evaluator

Information Gathering | Scenario Creation | Display Results

Sallie W. vs Luis P.

Judgment amount: $175.00
Filing fees: $180.00

Approximate time for judgment collection: 7 mos.

100%
Likelihood of Collection
0
3 6 9 12 15 18 21 24
Months

Simulations are displayed graphically, allowing the user to make comparisons among the different strategies. A timeline depicts how the likelihood of collection changes over time. Steps required to collect can be outlined for information purposes or to be followed as recommendations. Creditor expenditures, such as time lost from work, miles driven to court, and costs of filing are also estimated to aid in evaluating the costs/benefits of pursuit.

The **Enforcement Pursuit Evaluator** is an educational tool. Self-represented litigants unfamiliar with enforcement and collection issues can use it before beginning a lawsuit to learn about how the collection phase might develop. Self-represented litigants awarded a judgment can use it to help them assess the best way to pursue collection. Knowing the possible time, effort and cost of collecting a judgment ahead of time helps the litigant to make more informed decisions about pursuing a lawsuit and/or enforcing a judgment.

Scenario

Sallie's mom, Marge, is turning 75 in three months. Sallie really wants to do something special for her mother, but money is tight. She is barely making ends meet with her job at the grocery store. Marge's arthritis has been acting up and Sallie saw a warm paraffin spa tub specifically for people with arthritis that would be the perfect gift—luxurious and therapeutic. The paraffin tub costs $150 and, even saving a little every week, Sallie knows she won't be able to afford it.

In her spare time, Sallie helps people write resumes. She wrote a resume for her neighbor Luis, and he immediately got a new job and moved across town. He never paid her and still owes the agreed upon $175 for Sallie's time and effort.

When she calls Luis, he just says, "I got the job on my own. Your resume was worthless. I'm not paying." Sallie has been trying to get Luis to pay for six weeks and know that he'll never pay on his own. If she could just ge Luis to pay her, she could buy her mom a great present. She decides to sue him.

She doesn't know any lawyers, so she gets out the yellow pages and starts calling listings in her neighborhood. Sallie explains to a lawyer's receptionist what she wants to sue for, and the receptionist laughs and tells her that no lawyer would take her case. The receptionist suggests that she sue as a self-represented litigant and tells her to access the **Court Net** web site.

Sallie doesn't have a computer at home, so the next day at work she uses her work computer to visit the **Court Net** website. Sallie reads that sometimes cases take a long time to prosecute and that sometimes people aren't able to collect their judgments. She had no idea; Sallie always assumed civil cases were like the People's Court and that, when she won, Luis would hand her $175 in cash. The site recommends trying the **Enforcement Pursuit Evaluator** to see if litigation is a good idea for her.

The **Enforcement Pursuit Evaluator**, based on cases similar to hers, determines that she will have to spend $180 in filing fees, and that it would take approximately seven months to collect her judgment. The process guide goes on to show that she would likely have to file several motions to compel Luis to appear in court, and that she would have to take a lot of time off work in order to file and appear herself.

Looking at the graph of how long it might take, Sallie realizes that litigation might not be worth her time. She decides to see the best-case scenario, as it might be more encouraging. The best-case graph and steps are a little better, but suing would still require her to pay a filing fee up front, and she would have to take time off work. Sallie decides that her time is better spent doing other things. She will continue to pester Luis on her own, but will save her money for her mother's gift, rather than pay to file a lawsuit.

Form: 5/19/1998

Figure 23. Completing the **System Element** *is a Scenario that complements the static Discussion with an active description of the concept in operation.*

Scenario. Where the Discussion illuminates the structure of the System Element with regard to its essential components, the Scenario does the same thing for the way it works. The best static description never quite explains as well as following an example in operation. The Scenario employs that insight to provide a dynamic description. Expressed in present-tense style, the scenario delivers a user's-eye view of the System Element's features in action.

Conclusions

Generally speaking, two schools of thought exist on the structure of the planning and design process. In the simplest formulation of the traditional model, the process flows from analysis to synthesis to evaluation. More complex versions break down the three phases into substeps and introduce feedback loops, but the procedural dependence remains intact—analysis is done before synthesis, and synthesis is done before evaluation.

The conjectural/evaluative model challenges the lockstep relationship of the phases. In this version, ideas are generated and evaluated as they take form. Advantages are that ideas are less likely to be lost and that mistakes can be detected earlier. In a large project, this may mean avoiding massive redesign. To use this approach, however, there must be effective means of evaluation along the way. An appropriate model is the apprentice under continuous review by the master—the master not only reviews the work incrementally, but possesses the sum of experience and information necessary for judgment on a global as well as local basis. For a process to work in like fashion for a planning team acting as its own master, information should be explicit, available in detail, insightful enough to provide bases for both invention and evaluation, and richly cross-related.

Not coincidentally, the Structured Planning process has the means to take advantage of the conjectural/evaluative approach. First, there must be a way of knowing what to work on: the information base produced by Action Analysis provides that. Second, there must be a way to know whether an idea is contributing to a good solution: the Design Factors in the information base provide that at a local level, and the Defining Statements provide it at a global level. Third, there must be a mechanism to ensure that the planning team is not "climbing the wrong hill" in the parlance of optimization theory—creating piecemeal solutions that will be less than optimal once other Functions are considered. The structuring induced with the **RELATN** and **VTCON** programs reduces that danger significantly by tying together those Functions which ought to be considered concurrently.

The best approach to structure for the planning process, however, should use the best of both schools of thought. Good design philosophy refutes the folk adage, "You can't have your cake and eat it too"—in fact, creative thinking quite often finds a way to blend seemingly independent or even opposing ideas into a single, better solution. A perceptive planner tries never to be placed in the position of having to choose among goods; it is far better to think a bit harder and create one more alternative that integrates the best features of the competing choices. So, too, in this case.

The good in the traditional process model maximizes incubation time, holding off final ideas and evaluation of them until the last possible minute. As any planning or design project leader knows, more becomes known as the project proceeds, and the most is known at the end. The longer decisions can be responsibly delayed, the better is the chance that a more creative, higher-quality end result will be achieved.

The conjectural/evaluative model optimizes situational creativity, encouraging ideas when they occur and significantly reducing the likelihood that good ideas will be forgotten before they are considered "at the proper time". It also directs the progress of a project earlier because it encourages evaluation and, therefore, selection of ideas, as information is uncovered. Projects developed in this way are less likely to swing widely from concept to concept in later stages of synthesis.

Structured Planning draws from both models. Action Analysis dynamically juxtaposes discovery and invention in the creation of Design Factors, pressing early in the project for insights and ideas for how to use them. The virtues of the conjectural/evaluative early-action model are incorporated in that process. The strength of the traditional model appears when the information from Action Analysis, structured for optimal order of consideration, is finally arrayed for synthesis. The selection, modification and invention of ideas takes place then in an information environment rich in ideas—and steeped in the seasoning of incubation.

Planning and design are complex tasks. Products and systems can be made without good planning and design, but *excellent* products and systems cannot. Today, quality standards and development cycles do not permit the luxury of random success. The planning process must be *reliable* and *predictable;* reliable in that it can be depended upon to produce excellent concepts, predictable in that it can be expected to produce them on demand. Structured Planning is designed to meet those constraints.