EVOLUTIONARY STRUCTURED PLANNING

A Computer-Supported Methodology for the Conceptual Planning Process.

CHARLES BEZERRA AND CHARLES L. OWEN
Institute of Design
Illinois Institute of Technology, Chicago, Illinois, USA

Abstract. Complex design decisions require not only the use of information from the artifactual world (objective, quantitative data), but also from the world of culture (subjective, qualitative data). This paper describes Evolutionary Structured Planning, a computer-supported extension to a method that supports synthesis of objective and subjective information in the design planning process. With origins in Morphological Analysis and Structured Planning, the method introduces Genetic Algorithms as means for dealing with compound solution concepts in large combinatorial solution spaces. The method is extremely flexible and capable of being applied to virtually any design problem at the conceptual level.

1. Introduction

Technology today is particularly effective in providing understanding of how things work and how to incorporate that knowledge into products that can be industrially produced. As a somewhat unexpected result, the critical problem for design has shifted from "how to make products" to "what products to make". We probably already know how to make just about anything we can conceive; the problem now is what would be most useful to make? The crucial ingredient for conception, it is now being recognized, is better knowledge of potential products' users, those who will work with them, play with them, live with them, fix them – in sum, interact with products in all their modes of use. In contrast to the physical materials and phenomena incorporable in a product – which have measurable and predictable qualities – the actions and behavioral patterns of users are difficult to measure and quantify. Perhaps this is the reason for the lack of balance that seems to exist between the knowledge we have about how to make things and the knowledge we have about what to make.

"What to make" is an essential problem of design planning, an activity that takes place in the early phases of the design process. As input, design planning requires observations of user behavior. Ethnography, behavioral prototyping and protocol analysis are among several techniques now being perfected to meet this need. The deep understanding – insights – they can provide are key to the construction of creative, holistic solutions. Indeed, the sheer abundance of information that can be generated by these and other new analytic techniques mandates systematic organization in the planning process. For rational concept generation, synthesis and decisionmaking in these circumstances, there must be a systematic way to develop and process insights and ideas – especially when a design problem is complex enough that multiple agents are involved and the behavior of the system cannot be explained solely by an explanation of its parts' behavior.

This paper will discuss such a systematic, conceptual planning process, some of the tools already in use for systematic planning, and new developments for part of the synthesis section of the process.

Work in progress on the synthesis phase (giving rise to the title of this article: Evolutionary Structured Planning) will be described with a report on progress.

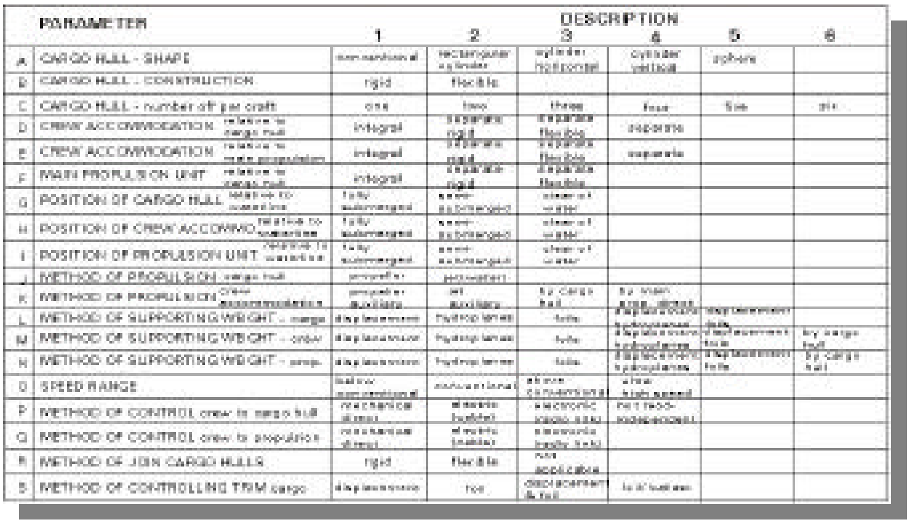
2. Background

Evolutionary Structured Planning fuses methodology from two longstanding conceptualization processes with techniques from a recently developed branch of artificial intelligence: genetic algorithms.

2.1 MORPHOLOGICAL ANALYSIS

The first conceptualization process, Morphological Analysis, has long been known in engineering design. Recognized for its success in the creative exploration of options for jet engine propulsion (Zwicky, 1947), Morphological Analysis was actually devised in the early years of World War II by Fritz Zwicky at California Institute of Technology. Since its publication, the method has been widely taught in engineering and product design and has been used broadly as a means for analyzing problems and identifying multi-faceted solution possibilities (Zwicky, 1969).

In its most direct form, the method constructs a "morphological chart" (Figure 1) that structures the problem to be addressed into a set of "parameters" or required functions listed as rows along the left side, and a set of "parameter steps" or alternative means for achieving the requirements of the parameters in the columns for each row. Parameters indicate what the solution must be, have or do; steps are options for individual parameter "solution". Solutions for the overall design problem are x-tuples found by selecting at least one step from each row (parameter) of the morphological chart.



	PARAMETER	DESCRIPTION 1	2	3	4	5	6
A	CARGO HULL - SHAPE	conventional	rectangular cylinder	cylinder horizontal	cylinder vertical	sphere	
B	CARGO HULL - CONSTRUCTION	rigid	flexible				
C	CARGO HULL - number off per craft	one	two	three	four	five	six
D	CREW ACCOMMODATION relative to cargo hull	integral	separate rigid	separate flexible	separate		
E	CREW ACCOMMODATION relative to main propulsion	integral	separate rigid	separate flexible	separate		
F	MAIN PROPULSION UNIT relative to cargo hull	integral	separate rigid	separate flexible			
G	POSITION OF CARGO HULL relative to waterline	fully submerged	semi-submerged	clear of water			
H	POSITION OF CREW ACCOMMOD. relative to waterline	fully submerged	semi-submerged	clear of water			
I	POSITION OF PROPULSION UNIT relative to waterline	fully submerged	semi-submerged	clear of water			
J	METHOD OF PROPULSION cargo hull	propeller	[illegible]				
K	METHOD OF PROPULSION crew accommodation	propeller auxiliary	jet auxiliary	by cargo hull	by main [illegible]		
L	METHOD OF SUPPORTING WEIGHT - cargo	displacement	hydroplanes	foils	displacement hydroplanes	displacement foils	
M	METHOD OF SUPPORTING WEIGHT - crew	displacement	hydroplanes	foils	displacement hydroplanes	displacement foils	by cargo hull
N	METHOD OF SUPPORTING WEIGHT - prop.	displacement	hydroplanes	foils	displacement hydroplanes	displacement foils	by cargo hull
O	SPEED RANGE	below conventional	conventional	above conventional	ultra high speed		
P	METHOD OF CONTROL crew to cargo hull	mechanical (direct)	electric (cable)	electronic (radio link)	[illegible]		
Q	METHOD OF CONTROL crew to propulsion	mechanical (direct)	electric (cable)	electronic (radio link)			
R	METHOD OF JOIN CARGO HULLS	rigid	flexible	not applicable			
S	METHOD OF CONTROLLING TRIM cargo	displacement	foil	displacement & foil	[illegible]		

Figure 1. A morphological chart for an ocean transport systems (Norris).

The morphological approach is a powerful technique for generating variety in solution, but it is difficult to use comprehensively for any but very small problem representations. The difficulty is evaluation.
Choosing among compound solutions requires comparative evaluation that quickly reaches exponential proportions. As an example, a simple 19 parameter problem set out as an illustration by Norris (1963) – with no more than 6 steps for any parameter (a total of 74 steps) – would require evaluation of over 6 billion possible solutions.

An only slightly more complex Norris example with 20 parameters and 88 steps required 9 x 4 x 3 x 6 x 3 x 3 x 6 x 3 x 3 x 6 x 3 x 6 x 3 x 3 x 6 x 3 x 3 x 6 x 3 x 6 = over 1.7 trillion evaluations. If each 20-step overall solution (1 step for each parameter) could be evaluated in a microsecond, it would still take over 20 full 24-hour days of computing to conduct all the evaluations. And this sized problem is not very complex by practical standards.

2.2 STRUCTURED PLANNING

The second conceptualization process is Structured Planning, a computer-supported planning process developed at the Institute of Design by author Charles Owen over a period of over thirty years (Owen, 1992, 1993a, 1993b, 1998) and under continuing development now (Bezerra and Owen, 1999). Structured Planning contains specialized information processing tools developed to help planners rigorously seek out, describe and structure information and, from the information structure developed, synthesize inventive, complex concepts that exhibit systemic properties.

Now implemented commercially, Structured Planning has been used for many projects at the Institute of Design over the years of its development, including a study of Space Station for NASA (1985), a prize-winning exploration of means for offsetting the effects of global warming (1990), a design for an ultra-large airship (1993), prospects for applications of molecular nanotechnology (1993), and most recently (1999), models for new forms of Internet and citizen/government network interaction. Student teams working with the process have won many awards including two Grand Prizes in the Osaka International Design Competition (1984, 1987) and the Grand Award for Environmental Technology in Popular Science's "The Year's 100 Greatest Achievements in Science and Technology" (1991). Steelcase and Kohler Company are among companies who are now using the process commercially for advanced product development.

Structured Planning can be regarded as a "toolkit" with methods that can be used independently or together, customized to the needs of a project. In its most general application, it proceeds through the following phases:

2.2.1. Metaplanning

From an initial idea for a project topic, research is conducted to understand the context the project would encounter, establish resources (people, money and time) to be used for the project, customize planning methodology, and establish major issues to be considered. The result is the assembly of an interprofessional project team and a charge to it in the form of a Charter with a project statement, background and guidelines for the project, a schedule, methodology and an initial list of key issues to be explored in defining the project.

2.2.2. Project Definition

With its Charter as "brief", the planning team begins its work with project definition. Issues from the topic list form nuclei around which policy discussions take place and research directions devolve.

Information developed is used to form arguable positions that are debated, attacked and defended until an insightful position statement can be written that will guide the actions of the team with regard to the issue. For each issue, a Defining Statement document is created (Figure 2) that summarizes the debate into a "question at issue", the team's arrived-upon position, alternative positions (not taken), and a discussion with background and arguments supporting the chosen position. The Defining Statements are a set of policy "white papers" defining the project in depth and providing the first set of criteria against which the system solution will be evaluated.

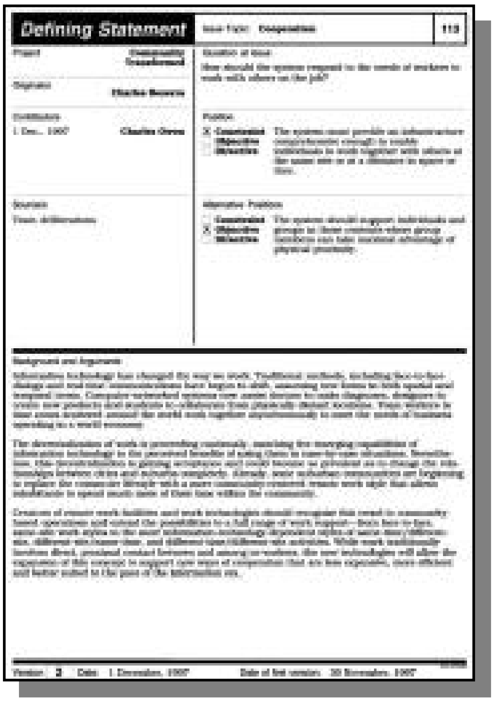


Figure 2. A Defining Statement for a community planning project.

2.2.3. *Information Development*

In this phase, an analysis of actions is undertaken to uncover Functions (what the system and its users must do), discover Design Factors (insights about the behavior of users and system), and invent Solution Elements (preliminary ideas for how to use the insights of the Design Factors to meet the requirements of the Functions). To optimize coverage of the project ground (as revealed by the Defining Statements) as well as to ensure that all users are considered, the team examines the life cycle of the projected system in detail employing a method of Activity Analysis. This analysis proceeds top-down from a consideration of major modes of behavior (for a transportation system, for example: site development, construction, operation, maintenance, repair, renovation, etc.), to specification of activities within each mode (within repair, for example: diagnosing, disassembling, making corrections, reassembling, testing), to description of the Functions performed in each activity (within disassembling, for example: check repair instructions, remove intervening parts, organize parts for reassembly, remove damaged parts, etc.).

The primary result of this analysis is a Function Structure that specifies the hundreds of Functions that must be performed well by the system and its many users (Figure 3).

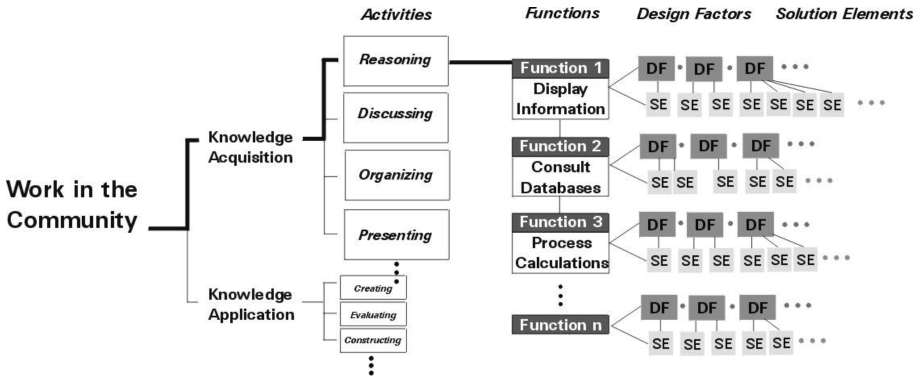


Figure 3. Developing a Function Structure.

In the process of conducting the Activity Analysis, planners also seek out information about what goes right or wrong in performing the Functions. These "insights" are written up as Design Factor documents (Figure 4), qualitative information that forms the backbone information base for the project.

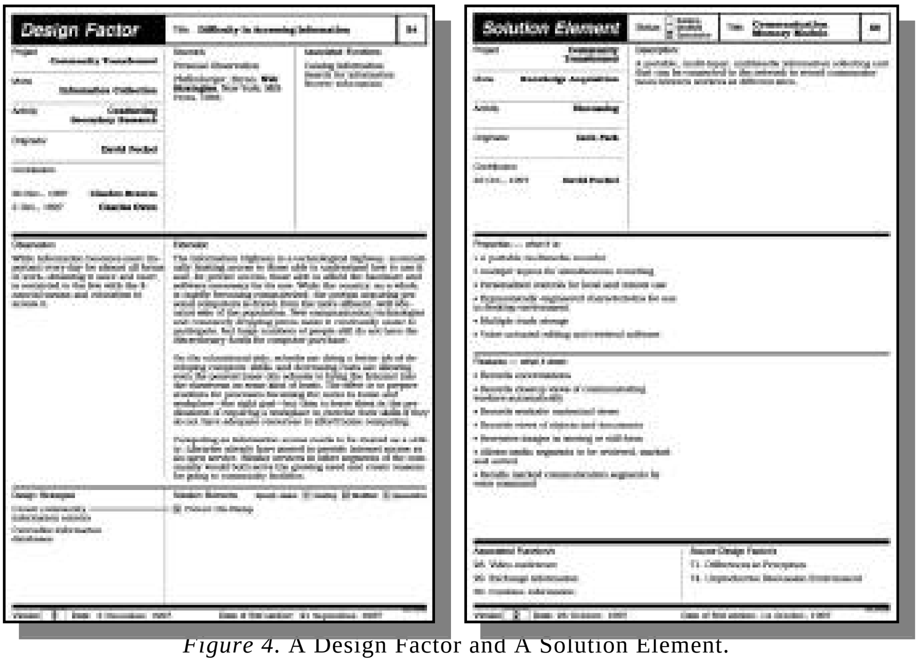


Figure 4. A Design Factor and A Solution Element.

Each Design Factor contains the kind of precious information that leads to good ideas for how to fix problems and take advantage of good user-system relationships. Also within the Design Factor documents are ideas for how to use the insights. This consciously exploits the reality that ideas come when they come; it is good policy to take advantage of insights with an on-the-spot appeal to the natural tendency to solve problems when they are perceived. Given evocative titles that ostentatiously represent their qualities, these ideas are listed as Solution Elements.

Separately, each Solution Element is given a short, one-page write-up of its own (Figure 4), capturing the idea in terms of a brief description, basic properties (what it is), and features (what it does).

However so sketchy, these Solution Element documents have enough detail to catch and hold ideas at the time when they are fresh and most fleeting. They become the initial set of ideas from which the final concept will metamorphose.

2.2.4. Structuring

The Function Structure, product of the top-down analysis process, serves its purpose elegantly in discovering the Functions to be fulfilled.

As a top-down process, it does a very thorough job of analyzing the system from the viewpoints of users across the system's life cycle. As a structure to guide the planning process, however, it has a serious flaw. Functions are together in the tree structure because they are derived from the same categories higher in the structure. If the Function structure were to be used to guide the synthesis process, concepts would naturally be developed back up through those categories, without regard for innovative possibilities that might be conceived from considering together Functions from other parts of the structure.

For this reason, a new structure is created in the structuring phase that puts Functions together, not because they share categorical derivation, but because they have potential solutions in common.

This structure, called an Information Structure (Figure 5), is created, bottom up, by two computer programs: RELATN and VTCON.

RELATN creates a non-directed graph of vertices and links, where each vertex is a Function and each link associates Functions that should be considered together because there are Solution Elements in relatively large number that either support fulfillment of both Functions or suggest potential conflict because there is support for one Function and obstruction for the other. In either case, the planner would wish to consider the two Functions together in developing component solutions.

As input to create the graph, RELATN takes an m x n matrix of n Solution Elements evaluated for m Functions.

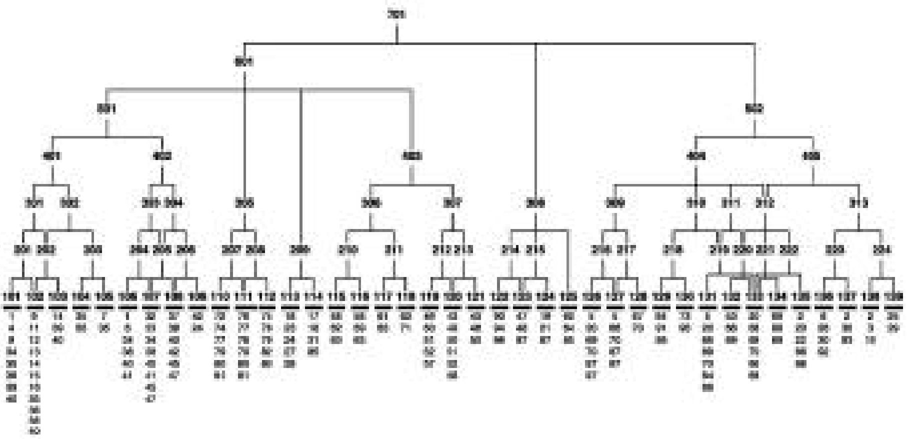


Figure 5. An Information Structure. 3-digit labels mark inclusion clusters. Functions are listed in primary clusters at bottom.

The Solution Elements created in the information development phase are scored against the Functions using the scale: +2 strongly supports fulfillment, +1 supports fulfillment, 0 No effect, -1 obstructs fulfillment, -2 strongly obstructs fulfillment.

VTCON decomposes the graph to find clusters of Functions that are highly interlinked, thus groups of Functions that should be considered together early in the synthesis process. Functions in these clusters typically will come from different parts of the Function Structure, brought together by commonality of solution ideas. From the initial decomposition, VTCON builds a hierarchical inclusion structure by progressively condensing together those clusters of Functions that have noticeable numbers of inter-cluster links or have Functions in common (Functions are allowed to be in more than one cluster – the original decomposition is not required to be a partition, and the hierarchical structure is then a semi-lattice rather than a tree). The final structure, usually several levels deep from its all-inclusive topmost single cluster to its primary level of the original decomposition clusters, is the Information Structure.

2.2.5. Concept Development

Working with the Information Structure as guide, the team selects, modifies, elaborates or creates new concepts to fulfill the Functions in the clusters. Means/Ends Analysis is a method useful in helping the team to understand the "meaning" of the clusters in the new structure, and Ends/Means Synthesis supports the synthesis of solution ideas, new and old, into a final set of System Elements. As the team works through each cluster, all the Solution Elements scored positively for its Functions in the structuring phase are available as viable ideas for the cluster. The synthesizing process moves from highly verbal discussions supported with rough sketches and diagrams to more defined descriptions and, finally, to formal presentations of refined ideas in a System Element format (Figure 6).

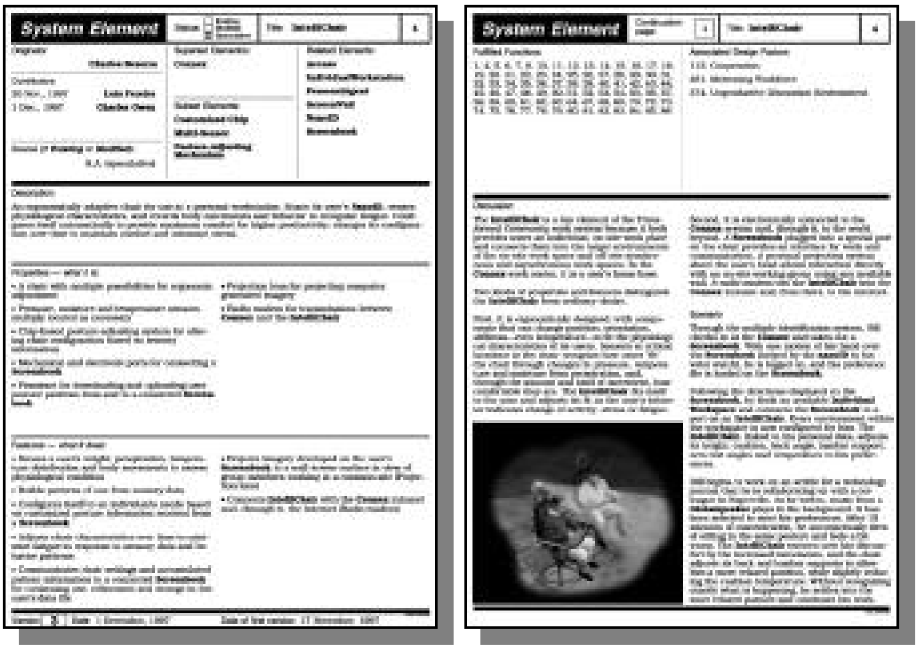


Figure 6. System Element documents.

2.2.6. Communication

In the final phase, a detailed write-up of the proposed system is prepared as a Plan with an Overview and a comprehensive set of System Elements. Each System Element has a description and discussion of properties and features as was prepared for the Solution Elements, but in addition, there is an extensive, illustrated discussion of the idea that includes any specific recommendations considered important by the planning team. The effect is to provide all the detail available without requiring its use by subsequent designing teams. The sections on properties and features specify what is necessary, the discussion section augments that with all that the planning team can add – for use by the designing teams if they can't develop something better.

Further, the System Element format includes a listing of all Functions fulfilled (for tracking the System Element's reasons for being), a listing of associated Design Factors (for checking generative insights), and a listing of Related System Elements (for navigating the overall solution of many System Elements).

Structured Planning continues to evolve as an effective planning process. In its present form, its analytic phases are very stable, the concept development phase is most open to experiment. A variety of tools have been used for synthesis – Means/Ends Analysis and Ends/Means Synthesis are most favored at present. Because for each cluster of Functions there are frequently large numbers of Solution Elements with something to offer, a process for identifying "best" combinations would make a significant contribution as the place to start the selecting/modifying/inventing final synthesis work. That identification process is the subject of this paper.

2.3. EVOLUTIONARY DESIGN

The third and last conceptualization process to be reviewed is what has become known as Evolutionary Design. Basically, Evolutionary Design is the use of techniques of evolutionary computation to generate designs. The process has its roots in computer science, evolutionary biology and design. Evolutionary computation, as a discipline of computer science, is concerned with search algorithms inspired by processes of natural evolution. These algorithms employ mechanisms analogous to the basic mechanisms of natural selection (mainly, selection, crossover and mutation) to evolve solutions to specific problems. Genetic Algorithms, or GAs, are the best known of these evolutionary search algorithms.

First introduced by John Holland (1975) in his research on natural and artificial systems, GAs are now being used in a variety of optimization procedures. GAs work with a population of individuals, where each individual consists of a phenotype and a corresponding genotype.

Phenotypes are characterized by a collection of characteristics or parameters; genotypes are the coded versions of these parameters. A coded parameter is normally referred to as a gene, with the values it can take being known as alleles. A collection of genes in one genotype is usually maintained internally as a string and referred to as a chromosome (Holland, 1975; Goldberg, 1989; Bentley, 1999).

In the evolutionary design process, an artificial version of "survival of the fittest" is implemented by a selection procedure that chooses individual solutions according to their "fitness", calculated by a fitness function; better solutions have a greater chance to survive and reproduce. As this selective process guides an informed search for better solutions, there is no need to systematically search the solution space.

3. Evolutionary Structured Planning

Evolutionary Structured Planning integrates these new computing concepts with the concepts of Morphological Analysis and Structured Planning.

Morphological Analysis has great strength as a method for analyzing functionality and developing multi-functional solution options.

Structured Planning is a sophisticated process combining a number of methods for devising evaluation criteria, uncovering functionality, recording insights and inventing solutions. Genetic Algorithms offer an elegant means for sidestepping the combinatorial limitations inherent in the need of both processes to find high-quality combinations of subsolutions.

Both Morphological Analysis and Structured Planning suggest a model for a conceptual design problem and its solution as a set of functions and a set of solutions, not necessarily mapped one-to-one, but mapped so that all functions are covered by one or more solutions.

This model corresponds directly to the phenotype/genotype/chromosome used by Genetic Algorithms. Structured Planning contains two evaluative systems that can contribute to the construction of the fitness function required by the Genetic Algorithm: Defining Statements as general criteria, and the Interaction Matrix created for the graph-making process employed in the RELATN program for specific Solution Element to Function assessment. The Interaction Matrix, required to build the Information Structure, already contains evaluations of all Solution Elements for all Functions with regard to support or obstruction. Finally, Morphological Analysis, in a form extended by author Charles Owen, incorporates a process of compatibility analysis to eliminate solution candidates that have internal incompatibilities among subsolutions. Used in Morphological Analysis to winnow the plethora of potential solutions, this technique can also be modified to increase the sophistication of the Genetic Algorithm's fitness function.

In the Evolutionary Structured Planning process described below, activities are interactive, combining the best of human insight and decisionmaking with the best of algorithmic computing. The process as described includes concepts from Morphological Analysis and Structured Planning as they are necessary to produce the Functions, evaluative information and functional organization, and new Genetic Algorithm processes as they can be incorporated to enhance synthesis in the concept development phase.

3.1. PHASE 1 - INPUT

The first task of Evolutionary Structured Planning is the identification of Functions and the development of Solution Elements. Input can be generated either by Morphological Analysis for simple problems or Structured Planning for more complex projects. If the generating process is Morphological Analysis, the first requirement is to describe a set of overall criteria with which to judge final solutions. The next step is to deconstruct the problem into a morphological chart with a well-balanced set of parameters or Functions and, for each Function, a set of steps or Solution Elements that covers as many possible alternatives as can be reasonably conceived. If the generating process is Structured Planning, the first step is to develop a set of Defining Statements that provides the overall criteria, and then to conduct an information development phase to identify Functions and Solution Elements (as described in Section 2.2.3).

Continuing through the phases, the Structured Planning team will arrive at the concept development phase with an Information Structure in which clusters at the primary level contain from a few to as many as 25-30 highly interrelated Functions. Taken one cluster at a time, sets of Functions from the Information Structure with their supporting Solution Elements are the subjects for evolutionary conceptual synthesis.

The computer program developed for experimenting with the evolutionary process incorporates an interface for this phase that helps the user to input Functions and Solution Elements, much as would be done in building a morphological chart. If input is from Structured Planning, the RELATN input file is used directly to provide the Function and Solution Element names. In either case, the final result of this phase is an initial data file that will be accessed by following phases.

3.2. PHASE 2 - SCORING (SOLUTION ELEMENTS VS. FUNCTIONS)

In the second phase, scoring is performed if it has not already been performed in the Structured Planning process. Following the model of Structured Planning, Solution Elements are assessed for their potential support or obstruction of Functions using the scale presented in Section 2.2.

Scales of any resolution may be used, but the five-value scale has been found to be most appropriate for the kinds of subjective evaluations that must be made in determining support or obstruction. Particularly high values: (+2) for support, or low values: (-2) for obstruction, will tend to increase the amount of interaction of a Function with other Functions, but only where the values are for the same Solution Elements.

3.3. PHASE 3 - SCORING (SOLUTION ELEMENTS VS. DEFINING STATEMENTS)

In phase 3, Solution elements are assessed for their potential support or obstruction of the overall criteria (Morphological Analysis input) or Defining Statements (Structured Planning input). At this level the criteria are general, applicative to overall problem or project goals – for example: cost, comfort, time to market, etc.

Each Defining Statement (or criterion) is assigned a weight to establish its importance as it is perceived for the overall design problem. Then, each Solution Element (or step) is evaluated for how much support or obstruction it affords the goals implicit in each Defining Statement; the scale used for assessment is the same scale (+2 to -2) used for assessing Solution Element vs. Function relationships. When the evaluation has been completed, the total score for each Solution Element is calculated as a weighted sum and saved for use in the fitness function.

3.4. PHASE 4 COMPATIBILITY ANALYSIS

The last analytical phase is concerned with discovering incompatibilities between Solution Elements. These occur inadvertently when ideas, good for the functions for which they have been independently conceived, are discovered to be unacceptable together. As a stark example, hydrofluoric acid is a good cleaning agent for some difficult metallic stains, and glass is a good low-cost container material for cleaning agents, but together they are highly discordant – hydrofluoric acid dissolves glass.

Simply eliminating all solutions with incompatibilities would help to reduce the combinatorial evaluation problem, but summarily eliminating solutions for incompatibility assumes that the incompatibility cannot be relieved or reduced to acceptable levels by a design team anxious to preserve qualities of the incompatible subsolutions in the final solution.

Another way to use incompatibility without resort to such draconian measures is to use it as a penalty factor – compound solutions with incompatibilities are assigned penalties that reduce their value relative to other solutions, but keep them in play to the extent that their other qualities continue to make them relatively desirable.

Incompatibility can also be graded. Some incompatibilities are mildly discordant, others are severe. Mild incompatibilities may be tolerated if there are other saving qualities, and even severe incompatibilities may be temporarily allowed in the hope that replacement of Solution Elements may lead to better overall solutions that are compatible. For evaluation, Solution Element pairings are judged for whether they are: 0 compatible, -1 discordant or -2 highly discordant. All Solution Elements are judged against each other in this phase with the results saved, as before, for use in the fitness function.

3.5. PHASE 5 - THE GENETIC ALGORITHM

The Genetic Algorithm used is a modification of the classic SGA, or Simple Genetic Algorithm (Goldberg, 1989). It contains non-binary string populations, roulette wheel selection, crossover, mutation and optimization mediated by a fitness function. The selection procedure represents a linear search through roulette wheel slots weighted in proportion to the values of the string. In other words, the chance of a solution being selected to reproduce is proportional to its fitness value.

3.5.1. Chromosome Representation

For Evolutionary Structured Planning, a chromosome represents an overall, compound solution to a design problem (Morphological Analysis) or an overall, compound solution to a part of the design problem identified by a cluster of Functions in the Information Structure (Structured Planning).

Genes in the chromosome are the Functions to be fulfilled, and their allele values are Solution Elements that work for the Functions. Thus, a chromosome is a string of integers, for example [3 1 9 2 9 2 14]. In this example, Solution Element 3 is the current choice for the first Function, Solution Element 1 for the second Function, 9 for the third, etc., and the entire string represents one complete, compound solution to the problem or partial problem represented by the 7 Functions. Figure 8 illustrates the structure.

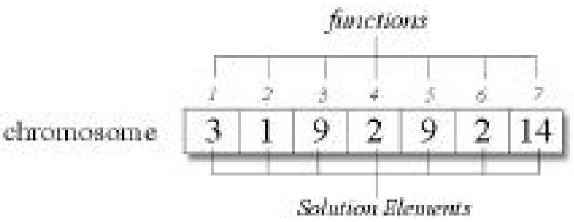


Figure 8. Chromosome representation.

3.5.2. Crossover and Mutation

Crossover, the fundamental process for producing offspring, is performed with a one-cut-point crossover operator (Goldberg, 1989) as follows: for a pair of selected strings an integer position *p* along the string is randomly selected between 1 and the string length *q* less one [1, *q* - 1]. As shown in Figure 9, two new strings are created by simple exchange of all characters between positions *p* + 1 and *q* inclusively.

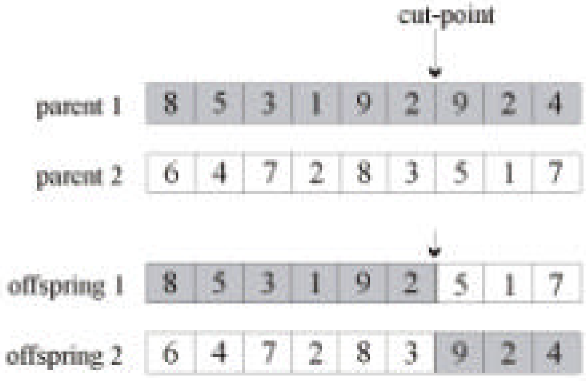


Figure 9. Crossover operation.

The second generative process, mutation, is performed as a random perturbation of one of the genes within the chromosome. A gene (Function) is chosen at random, and its allele (Solution Element) is randomly changed to another from the set of all its supporting Solution Elements. An example of the mutation operation is shown in Figure 10.

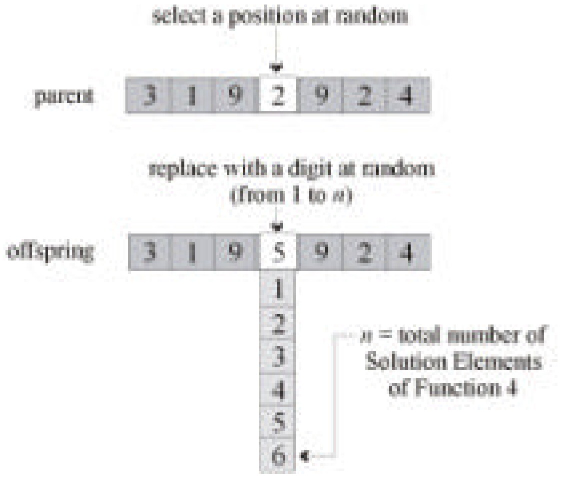


Figure 10. Mutation operation.

3.5.4. Fitness Measure

The Fitness Measure determines which offspring (and parents) survive each round of evolution. For Evolutionary Structured Planning, the measure combines information from the scoring operations of phases 2 and 3 with a penalty function from phase 4. Weights for each of the three parts of the measure are also incorporated to allow experimentation with the relative importance of the three types of control.

$$F(x) = \frac{W_1}{2n^2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} + \frac{W_2}{2n \cdot \sum_{k=1}^{m} w_k} \cdot \sum_{i=1}^{n} \sum_{k=1}^{m} w_k \cdot B_{i,k} + \frac{W_3}{n(n-1)} \cdot \sum_{i=1}^{n-1} \sum_{j=2}^{n} C_{i,j} \qquad (1)$$

Where:

x = compound conceptual solution (chromosome)
F(x) = fitness of solution
m = number of Defining Statements or overall criteria for the project or problem
n = number of gene positions in chromosome (Functions for phenotype, Solution Elements for genotype)
i = index for gene position within chromosome
j = index for gene position within chromosome
k = index for Defining Statements
Aij = evaluation for Solution Element i vs. Function j (from Phase 3)
Bik = evaluation for Solution Element i vs. Defining Statement k or criterion k (from Phase 2)

C_{ij} = penalty for incompatibility between Solution Elements i and j (from Phase 4 – value will be 0 or negative)
W_1 = weight given to quality of overall solution vs. Functions
W_2 = weight given to quality of overall solution vs. Defining Statements
W_3 = weight given to compatibility of Solution Elements
w_k = weight given to Defining Statement k or criterion k

By varying the weights W, different levels of importance can be placed on the three forms of evaluation. It is anticipated that this will influence both the rate and direction of evolution.

4. Experiments and Preliminary Results

A program to test the evolutionary algorithm has been developed in the object Pascal programming language, and a number of test runs have been made to confirm the effectiveness of the evolutionary approach to concept construction. Although experiments and refinements are still in progress, some preliminary comments can be made on insights gained thus far.

First, it is very important to emphasize that design planning is not optimization. This early in the design activity the process is more exploration than search, more about raising possibilities than eliminating them. Definition and understanding of the problem are evolving alongside ideas for its solution. Thus the goal of an evolutionary exploration process should not be to find singular "best" compound solutions (optimization), but to find populations of insightful combinations for the stimulation of creative synthesis.

As a problem may have more than one solution, a Function can have more than one Solution Element. During the process of conceptualization, new Solution Elements (final System Elements) should be creatable by combination or modification of existent ones. By bringing interesting seed combinations to the attention of the planning team, the evolutionary methodology appears to be supportive of this kind of transformation and combination.

An effective computer-supported method should also protect the planning or design team from being forced into the position of having to choose among "goods". Instead of leading the process toward a choice among different candidate designs, it should support a strategy of "having your cake and eating it too", helping the team to find or create that "one more" alternative that integrates the best features of the competing choices, thus avoiding the problem of choice. The subtle as well as dramatic variations present in the candidate populations the process presents work well toward this goal, frequently pointing the way to combinations that can be modified to best advantage.

To test the performance of the program, initial experiments on dummy design data were carried out using a high crossover rate and low mutation rate with a relatively small population of 50 individuals (chromosomes).

Design problems were limited to a maximum of 30 Functions (i.e., chromosome strings of maximum 30 genes), and the number of Solution Elements (alleles) for each Function were restricted to counts between 1 to 30. The compound solutions generated by these initial experiments thus occupied a 30 x 30 morphological chart, large enough to test a very large combinatorial solution space (worst case with 30 Solution Element options for each Function: 30 x 30 x ... x 30 = 30^{30} compound solutions).

Academic and commercial applications of Structured Planning methodology (such as noted in section 2) suggest that system design problems of the sort effectively addressed by Structured Planning usually vary between 150 and 250 Functions, and Functions seldom have more than 30 Solution Elements that are scored positively for them from the entire set of Solution Elements for a project – and considerably less that are intentionally developed for them. In the conceptualization phase of Structured Planning, the Information Structure guides the development of final compound solutions of System Elements, and the number of Functions in any primary level cluster is almost never more than 25. The 30 x 30 restrictions of the experimental model are therefore quite in line with what can expected in actual use.

Computer runs using a Pentium 300 MHz computer running the Windows NT operating system required roughly 1 hour and 40 minutes to evolve populations of high quality solutions. Considerable variation in time was expected (and occurred) as data variations were introduced, but the execution time was well within acceptable limits. Figure 11 graphically shows the typical convergence process. In fact, considerably longer times are tolerable as long as the process is not required to be interactive in any real-time sense, and the Structured Planning process would incorporate the evolutionary process most readily as a preparation process for the final conceptualization phase – not as an active, interactive process.

The evaluations conducted in the scoring phases are time consuming but vitally necessary activities in the process. The values generated are required for the evolution of candidate solutions, of course, but the act of evaluation has great additional hidden benefit. In the process of deciding how a Solution Element may support or obstruct a Defining Statement or Function, meaningful new understanding of the Solution Element grows. The many viewpoints required for the Solution Element to be considered over and over again with different Defining Statements and Functions add layers of new detail to the concept. Moreover, new meaningful ideas and relations are often discovered that may lead to significant changes or additions to the Solution Elements. Despite the use of interfaces specifically designed to reduce the time spent in the evaluation process, it remains a time consuming activity, although one with redeeming reward.

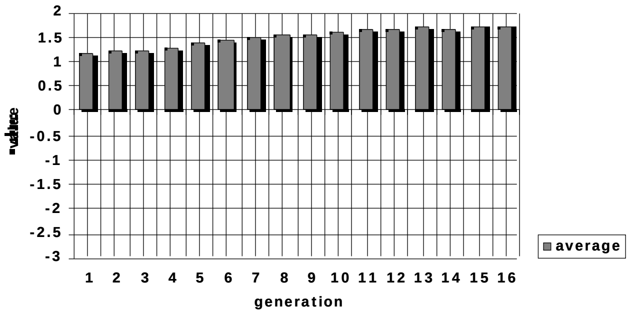


Figure 11. Evolution of quality for a population of compound solutions. Maximum normalized value for unweighted fitness function is 2.0.

The fitness measure is designed for experiment with differential weighting of the three forms of solution evaluation. In tests where the weighting allowed only the first two terms (evaluations against Functions and Defining Statements), the algorithm converged readily to populations with high scoring Solution Elements. However, when the third term was included (bringing in compatibility analysis), time to convergence rose, and convergence to high scoring solutions was neither as direct nor as complete. Further experiments here are indicated; some of the directions planned are discussed in the next section.

When included, compatibility is addressed by a penalty technique; the third term is zero unless incompatibilities are detected, and then is always negative. In genetic algorithms, the penalty technique is used to keep a certain number of infeasible solutions in each generation to force the genetic search process toward an optimal solution from both regions of feasibility and infeasibility (Gen and Cheng, 1997). In other words, incompatible solutions are not rejected out of hand because some may guide the search toward a most promising area of the solution space that might not be found from the other side.

5. Conclusions and Future Research

Joining combinatorial optimization techniques with design planning is not an obvious strategy to follow, because optimality is generally regarded in design planning to be a quality sought later in the development process. Complexity, however, is a catalyst now forcing a change in view. Complex problems invariably seem to require complex solutions, and modern systematic planning methods readily reveal many more options to explore than can easily be processed with conventional methods. Evolutionary Structured Planning weds evolutionary computation techniques to contemporary planning processes to equip them for combinatorial-scale explorations. The result offers significant help to the planning team hoping to use its limited time wisely.

The work completed thus far demonstrates in principle that evolutionary computational processes have value for conceptual level planning and design. Further research opportunities exist. Besides continuing experiments and refinements with specialized test problems and case studies observing actual project teams working on real problems, possibilities for study include making more modifications – specific to the design-planning process – to solution (chromosome) description, the offspring generating process, and the evaluative process. Two examples of the latter are currently being formulated:

1. Compatibility analysis thus far has been limited to its negative side – incompatibility. Incompatibility also has its obverse; Solution Elements may be very compatible – actually working better because of the synergistic presence of a catalytic or symbiotic Solution Element. For this reason, a full compatibility/incompatibility scale could be used to judge this quality, and the "penalty" factor for discordance reversed to a "reward" factor when the situation warrants. For evaluation, Solution Elements would then be judged for whether they are: +2 highly synergistic, +1 synergistic, 0 compatible, -1 discordant or -2 highly discordant, with the results saved, as before, for use in the fitness function.
2. The current fitness function term for the quality of Solution Elements vs. Functions considers only scores for Solution Elements in the chromosome vs. the Functions represented in the chromosome. Structured Planning, however, produces (as a byproduct of its structuring process) scores for all Solution Elements vs. all Functions, system-wide. A measure of a Solution Element's quality across the system, thus, could be easily used in place of or in combination with its chromosome-specific value.

Acknowledgements

This research has been supported by the CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil, and by the Institute of Design, Illinois Institute of Technology.

References

Bentley, P. J.: 1999, *Evolutionary Design by Computers*, Morgan Kaufmann, San Francisco.

Bezerra, C. and Owen, C. L.: 1999, Managing complexity in design: the role of computer-supported methods, *in Proceedings of The 15th International Conference on Computer-Aided Production Engineering*, University of Durham, UK, pp. 3-8.

Gen, M. and Cheng, R.: 1997, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York.

Goldberg, D. E.: 1989, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts.

Holland, J. H.: 1975, *Adaptation in Natural and Artificial Systems*, (4th ed.) University of Michigan Press, Ann Arbor, Michigan.

Norris, K. W.: 1963, The morphological approach to engineering design, in J. C. Jones and D. G. Thornley (eds), *Conference on Design Methods*, Macmillan, New York, pp. 115-140.

Owen, C. L.: 1992, Context for creativity, *Design Studies*, **13** (3), 216-228.

Owen, C. L.: 1993 A critical role for design technology, *Design Management Journal*, **3** (2), 10-18.

Owen, C. L.: 1993, *Design for Integrity*, Institute of Design, Illinois Institute of Technology, Chicago.

Owen, C. L.: 1998, Design, advanced planning and product development, in A. de Moraes and A. Jefferson de Oliveira (eds), *P&D Design 98*, Vol. 1, Proceedings of the 3o Congresso Brasileiro de Pesquisa e Desenvolvimento em Design, Associacao de Ensino de Design do Brasil / Revista Estudos em Design, Rio de Janeiro, pp. 41-56.

Zwicky, F.: 1947, Morphology and nomenclature of jet engines, *Aeronautical Engineering Review*, **6** (6).

Zwicky, F.: 1969, *Discovery, Invention, Research through the Morphological Approach*, Macmillan, New York.