A Systems Engineering Approach to the Concept Development Process

by

Mark Zvi Wilen

S.B. Mechanical Engineering, Massachusetts Institute of Technology

Submitted to the Sloan School of Management
and the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Management
and
Master of Science in Mechanical Engineering

In conjunction with the Leaders for Manufacturing Program
at the Massachusetts Institute of Technology
May 1995

Signature of Author ______________________________
Department of Mechanical Engineering
MIT Sloan School of Management
May 24, 1995

Certified by ______________________________
Woodie C. Flowers
Pappalardo Professor of Mechanical Engineering
Thesis Advisor

Certified by ______________________________
Karl T. Ulrich
Associate Professor of Management Science
Thesis Advisor

Accepted by ______________________________
Jeffrey A. Barks
Associate Dean, Master's and Bachelor's Programs
MIT Sloan School of Management

A Systems Engineering Approach to the Concept Development Process

by
Mark Zvi Wilen

Submitted to the Sloan School of Management
and the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degrees of
Master of Science in Management and
Master of Science in Mechanical Engineering

Abstract

It is the goal of this thesis to propose techniques for improving the design process as practiced in developing complex manufactured products. The process presented here is a guideline for generating and recording the necessary information to carry out the concept development process. This thesis includes a presentation of the background, an explanation of the underlying philosophies, a description of the process steps, a discussion of the benefits, and recommendations for further research.

This process was developed through process simulation activity conducted at The Boeing Company's Large Airplane Development division and is based on the principles of systems engineering. The principles led to a function-driven, top-down, and system-wide-focus approach to concept development. The primary benefits of this approach include optimizing the total system, a more efficient process and product, and a better means for handling change.

The proposed concept development process builds on existing research in the field of product development and systems engineering. It is in the bringing together of these two disciplines that this process ventures into new ground. In addition, several innovations were made during the process simulation activity. These innovations and other aspects of the process enable more information to be considered during up front decisions in order to take into account the impact on the total product and its associated design, build, support, and business processes. In this way, the overall product, instead of the individual elements, will be optimized.

A possible implication of the process presented in this thesis is that this approach can be successfully extended beyond concept development into detailed design. However, further research will be required to confirm this hypothesis.

Thesis Advisors:

Woodie C. Flowers
Pappalardo Professor of Mechanical Engineering

Karl T. Ulrich
Associate Professor of Management Science

Acknowledgments

I wish to acknowledge the Leaders for Manufacturing Program for its support of this work and for providing me the opportunity to vastly expand my mental model.

I wish to thank Lindsay Anderson for developing and supervising this project and for kindly helping me navigate the channels within The Boeing Company.

I would like to thank my advisors Woodie Flowers and Karl Ulrich for their inspiration, guidance, support and most of all for their love of design.

I must thank The Boeing Company and the Large Airplane Development team for their gracious hospitality and for providing the learning laboratory to develop the processes presented in this paper. I would also like to thank several individuals within Large Airplane Development who helped make this work possible. As members of the Process Development Support Team, Robert Kelley-Wickemeyer, Gill Thompson, Sue Stone-Byrne, and Jim Barry were enormously helpful and encouraging. Furthermore, the people I worked with on teams contributed by bringing years of design experience and breathing life into this project. They include Peter Colebrook, Mike Reberg, Dave Sommer, Marian Luca, Cheryl Darling, Walt Howard, Doug Copeland, Sarah Ebner, John Munroe, Wade Meyer, Gib Gadeholt, Paul Phillips, Ben Almojuela, Jim Nisbet, Bob Fulford, and Jack Kleiber.

Finally, I would like to thank my family for their wonderful support and my friends who helped me take time out to smell the flowers and hear the music during the LFM experience.

Contents

I. Introduction

I-A) Background

> Among intellectual and technological pursuits, design is one of the oldest endeavors. From prehistoric periods, humanity has designed and made hunting implements, shelters, and clothing. It might have preceded the development of natural sciences by scores of centuries. Yet, to this day, design is being done intuitively as an art. It is one of the few areas where experience is more important than formal education.

These words, which open Nam P. Suh's *The Principles of Design*, reflect the struggle many organizations face in striving to develop a more reliable and transferable process for product design. A purely intellectual and technological approach to product design is by no means advocated here: indeed, great design is achieved in the combination of both art and science. However, there are techniques to improve the analytical aspects of the design process without dispelling its magic. Furthermore, experience is an invaluable ingredient in product design; yet there are techniques to help a design team climb the learning curve more rapidly. It is the goal of this thesis to propose techniques for improving the design process as practiced in developing complex manufactured products.

The techniques for improving the design process proposed here were developed specifically for the concept development process. The process was developed through process simulation activity conducted at The Boeing Company's Large Airplane Development division and is based on the principles of systems engineering. In applying a systems engineering approach to concept development, this process optimizes the total system, results in a more efficient process and product, and better handles change.

I-B) Underlying Philosophy

The process described in this thesis rests upon an underlying philosophy based on the following principles of systems engineering.

- Function-Driven
- Top-Down
- System-Wide Focus

Combined, they embody a systems engineering approach to the concept development process, an approach that is not commonly applied at the outset of a design project. *Function-driven* signifies that the product functionality is developed before the physical design is configured. It can be thought of as understanding the problem before jumping to the solution. *Top-down* is both an orderly method for developing product functionality as well as an efficient approach to problem solving. *System-wide focus* is applied to draw attention back to the core problem. The drawback in designing a complex product through a hierarchical process is that once the problem is broken down into simpler subproblems and then solved, the pieces don't always fit back together. Therefore, we repeatedly apply a system-wide focus during this process to ensure that while breaking down the problem we are still optimizing the solution for the whole, and not just for the individual parts.

I-C) Benefits of the Process

This approach to concept development provides the following benefits.

- *Optimizes the 'Total System:'* The process includes several steps which focus the design team on the big picture, influencing decisions to optimize the overall functional system.

• *More Efficient Process & Product:* The top-down approach provides a more efficient process which also leads to a more efficient product as well as a higher quality product.

• *Better Handles Change:* Change can be handled more competently due to information on the fundamental and incidental functional interactions captured during this process. This applies both to changes arising during the course of designing the product and to future changes made after the product has been released.

• *Provides an Organizing Mechanism:* The function tree created during this process provides a means for organizing all the information generated during the concept development process as well as a means for organizing the human resources required to perform product development.

I-D) The Process

The concept development process proposed in this thesis builds on existing research in the field of product development and systems engineering. It is in the bringing together of these two disciplines that this process ventures into new ground. In addition, several innovations were made during the process simulation activity. The process should be applied in the spirit that it was developed, by continually seeking ways to improve it and questioning its limitations.

This process also provides techniques for incorporating more information into up front decisions. This information enables the design to be shaped in conjunction with anticipated future variability and to take into account the impact of the proposed product concept on the associated design, build, support, and business processes. In providing tools to efficiently organize this information, this process promotes better decision making and should result in fewer costly design iterations.

Therefore, the process presented here should be viewed as a guideline for generating and recording the necessary information to carry out the concept development process; not as a recipe to be followed without questioning. In no way is this meant to serve as a replacement for creative, independent thought. It is, however, designed to serve as a means for channeling individual and team creativity towards more productive ends. Finally, the goal is not to complete these steps: it is to generate and synthesize the data required to make the right design decisions and to record them for future reference.

I-E) Thesis Contents

In the following chapters the background will be presented, the underlying philosophies will be explained, the steps of the process will be described, the benefits will be discussed, and finally topics for further research will be suggested.

II. Background

II-A) Internship at The Boeing Company

The Leaders for Manufacturing Program augments the academic experience with a six month research internship at a manufacturing firm. My internship brought me to the Large Airplane Development team at The Boeing Company, where I participated in the process development activities they were conducting.

Process Development Activities

When I arrived, Large Airplane Development (LAD) was faced with challenge of developing a 600 to 800 passenger airplane. The proposed product will thus be capable of carrying between 50% and 100% more passengers than the largest product Boeing sells today, the recently introduced 747-400. Furthermore, this airplane will be required to provide substantial savings to its airlines customers, in both operating and purchasing costs, in comparison to the best alternative, again the 747-400.

While improvements of five percent to ten percent can be achieved through incremental progress, more substantial improvements require fundamental changes. Therefore, in order to meet these challenges, LAD embarked on a mission to fundamentally rethink their process for developing, producing, and servicing an airplane product. This *Process Development* effort began with several focused teams, each one concentrating on a specific group of activities.

For the first few months of my internship, I joined a team of ten Boeing engineers and planners to rethink the concept development aspect of the design process. Our task was to define *how the concept development process should be.* By the time the team was finished, nearly 8000 person-hours had been devoted to developing a new paradigm for the concept development process. This new paradigm could be thought of, in reduced form, as the three underlying philosophies described in the introduction.

Before the team fully completed their charter, I began independent research into techniques and methods in order to apply the concept development approach developed by the team. In doing so, I reviewed the product development research of Karl Ulrich, Don Clausing, and Kim Clark, among others. I was also encouraged by my academic advisors to look into the literature on systems engineering. After a few weeks of research, I had amassed several techniques into some semblance of this new concept development process. In order to prove its worth, the Process Development Support Team (which was supervising the process development activities at LAD), suggested I conduct a process simulation.

Process Simulation Activities

Rather than conduct an actual design exercise with engineering models and CAD drawings, a process simulation entails working through a process with the *type* information needed to make the *types* of decisions required. During the simulation, we would ask ourselves what decisions need to be made now in order to carry the design further, with minimal risk of having to go back to make revisions. Then we would ask ourselves what kind of information would be required to make those decisions. We would generate reasonable approximations of that information and then try to make decisions based on those approximations. Usually, we would know right away whether we had enough of the right information to make a decision. We were greatly assisted in this process by the many years of design experience collectively possessed by the team.

As the simulation process continued, it would become apparent if some information, or certain decisions, were omitted. In those cases we would go back and revise the process, to correct for the deficiencies. The process of correction did not just involve adding the missing elements, but questioning the structure of our approach to see if it was still sound given this new knowledge. In trying to match our methodology to our underlying philosophy, we were careful not to tweak the process without understanding why it was broken. In this way we brought a continuous improvement approach to process development. In fact, Figure 4.1, which describes our process development process, resembles the Plan-Do-Check-Act Cycle (or a PDCA Cycle) at the basis of continuous improvement activities.

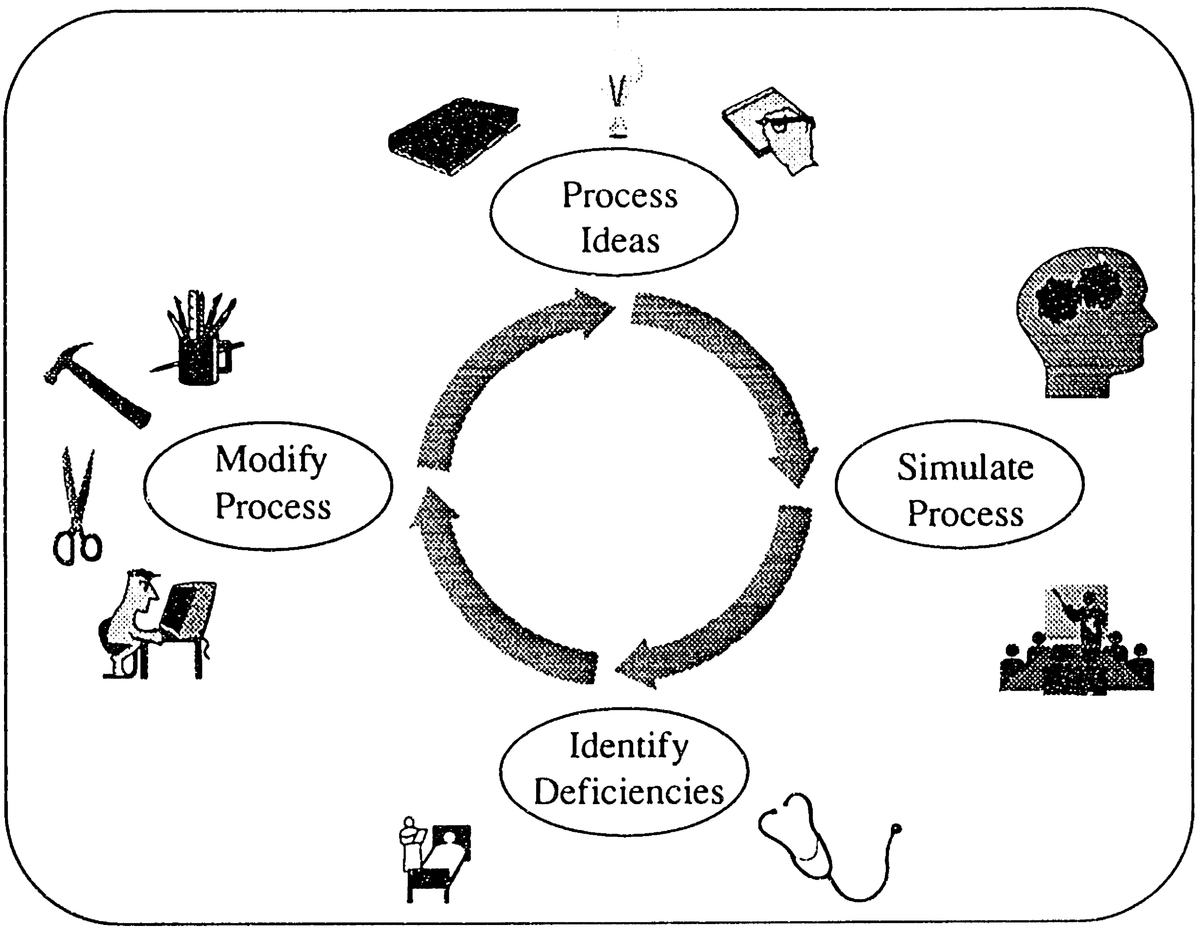

Figure 2.1: Process Simulation Cycle

The first process simulation lasted one week, by which time it became clear that there were some serious deficiencies in the process. We realized we were lacking a methodology to iterate between the function tree and the physical element tree. Without such a technique, our function-driven approach was unmanageable. Therefore, it was back to the drawing-board, or to the library as the case may be. The missing link was found in a research paper written by Sontow & Clausing (sent by Don Clausing after we consulted with him). After modifying the process to include this new approach for iterating between the function tree and the physical element tree, a second process simulation was conducted. In this second effort, six engineers spanning a range of disciplines and one cost estimator with several years of manufacturing experience "simulated" for about two and a half weeks. The modified process was much more effective, enabling us to develop seven layers of both the function tree and the physical element tree (these can be seen in Figure 2.2 and Figure 2.3). During that second simulation, we also made incremental improvements as minor deficiencies were recognized. The resulting process steps are described in Chapter IV.

II-B) Concept Development Process

A little bit of background is required to explain how concept development is defined for the purposes of this thesis. First of all, the scope of product development considered here is that of complex manufacture products, or products consisting of 10,000 to 100,000 unique parts (i.e. aircraft, automobiles, etc.). Therefore, when we compare the processes presented in this thesis to how concept development is typically carried out, we are using the practices employed in the mid 1990's by large US manufacturing firms of complex products as a reference.

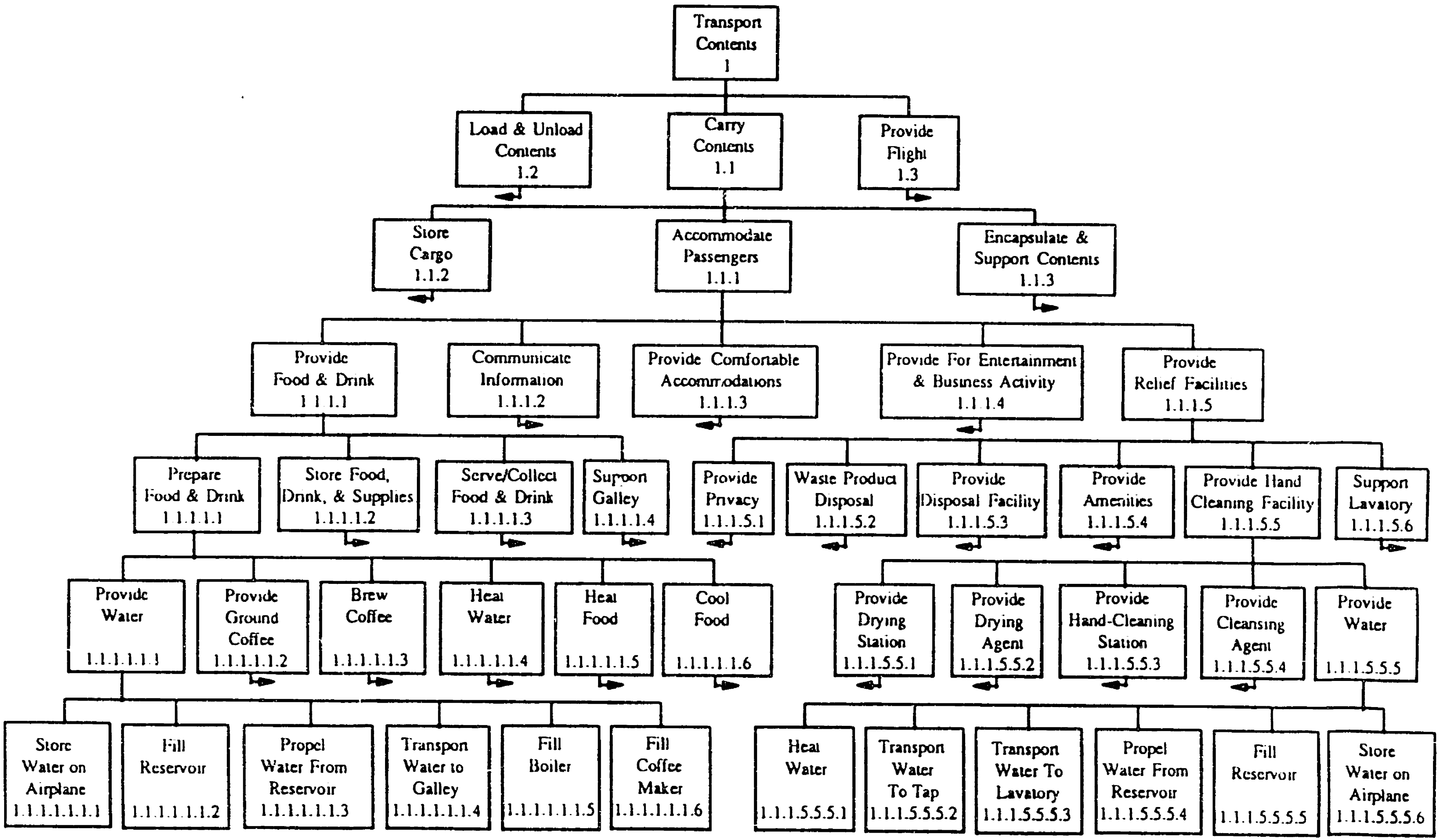

Figure 2.2: Function Tree Developed During Simulation Activity

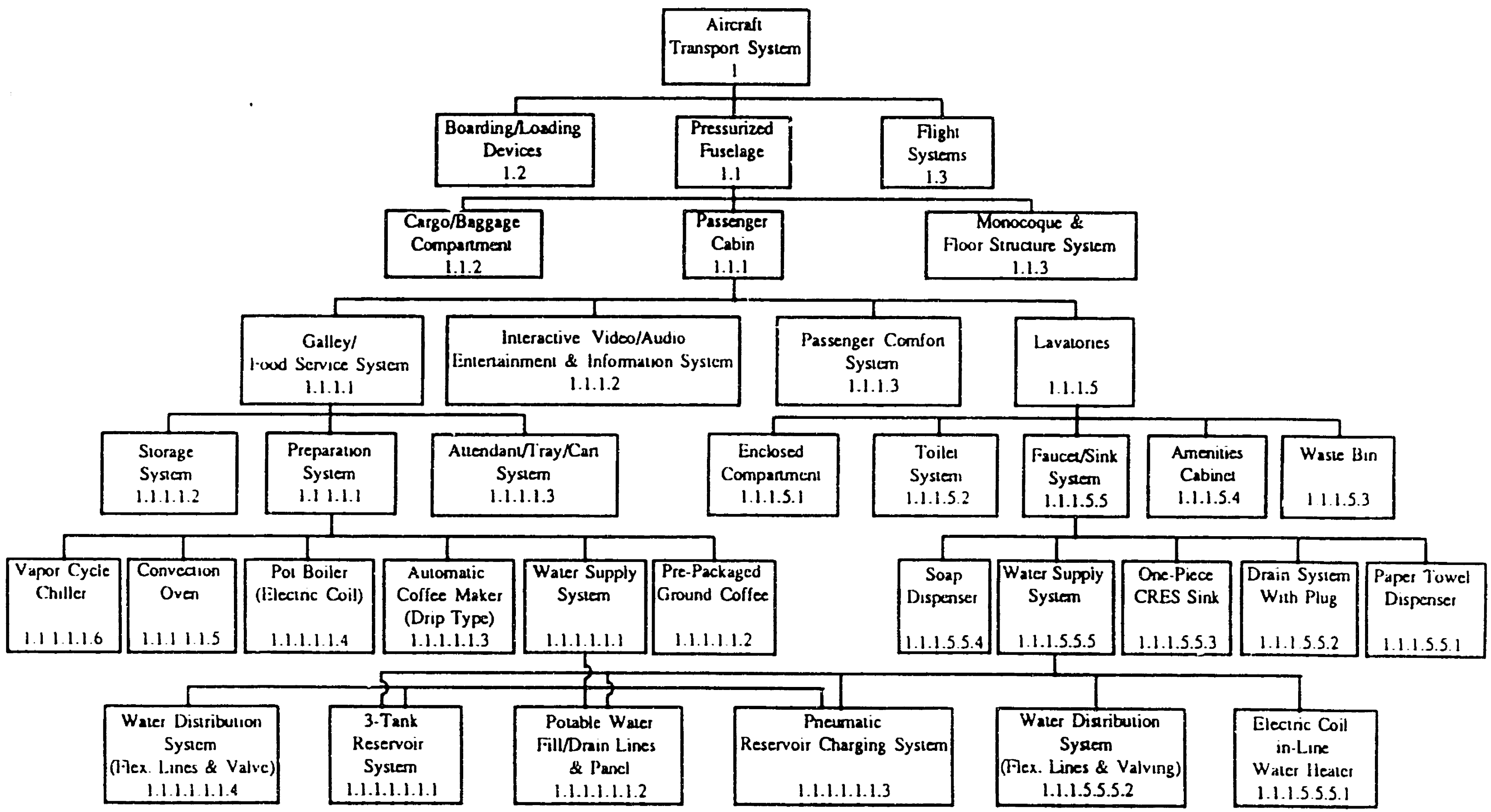

Figure 2.3: Physical Element Tree Developed During Simulation Activity

For the purposes of this thesis, the *concept development process* is defined as the process through which several product concepts are generated and the best one is selected.[1] Concept development is thus the first step of the design process, which progresses through system development and finally to detailed definition, as depicted in Figure 2.4. A possible implication of the process presented in this thesis is that this breakdown may not be necessary. It might be possible for system development, and even for detailed definition, to be carried out through the same process as concept development. However, further research is required to confirm this hypothesis.

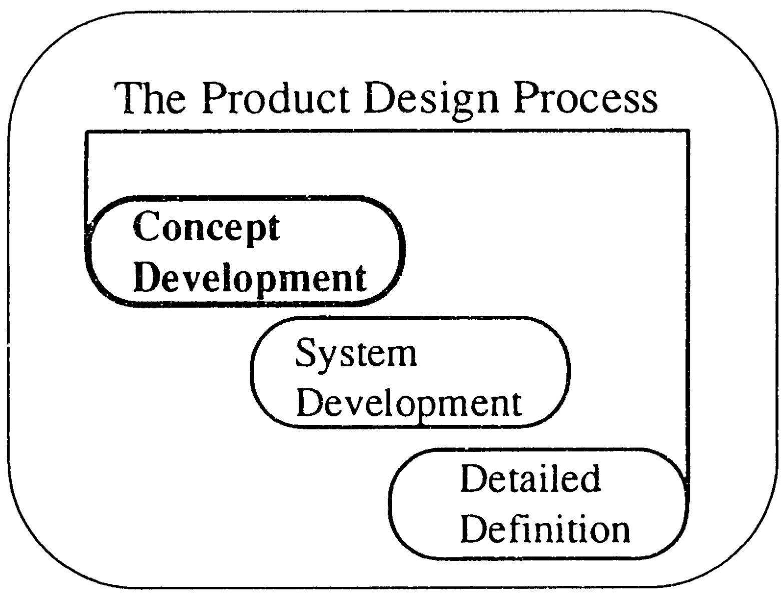

Figure 2.4: Steps of the Product Design Process

This thesis focuses on product development in a manufacturing firm where market research and product strategy preprogram activity identifies a niche in the marketplace not being met by the company's existing product line. The market research and product strategy activities then collect the customer requirements and identify competitive

[1] It should be noted that this is a rather narrow definition of the concept development process, which is often considered to include identifying the needs of a target market, benchmarking competitive entries, etc. (Ulrich and Eppinger, 1995).

challenges for that target market. This information is used to create a mission statement which communicates what the product will provide the target customers and how it will further the company's strategic goals. The concept development process is then initiated by a management decision to pursue the product proposed in this mission statement.

Figure 2.5 portrays these organizational influences on product development. It further depicts the constraining influences on product development imposed by government regulatory agencies, research and development, manufacturing, and partners and suppliers. It is within this product development framework that the concept development process proposed in this thesis operates.

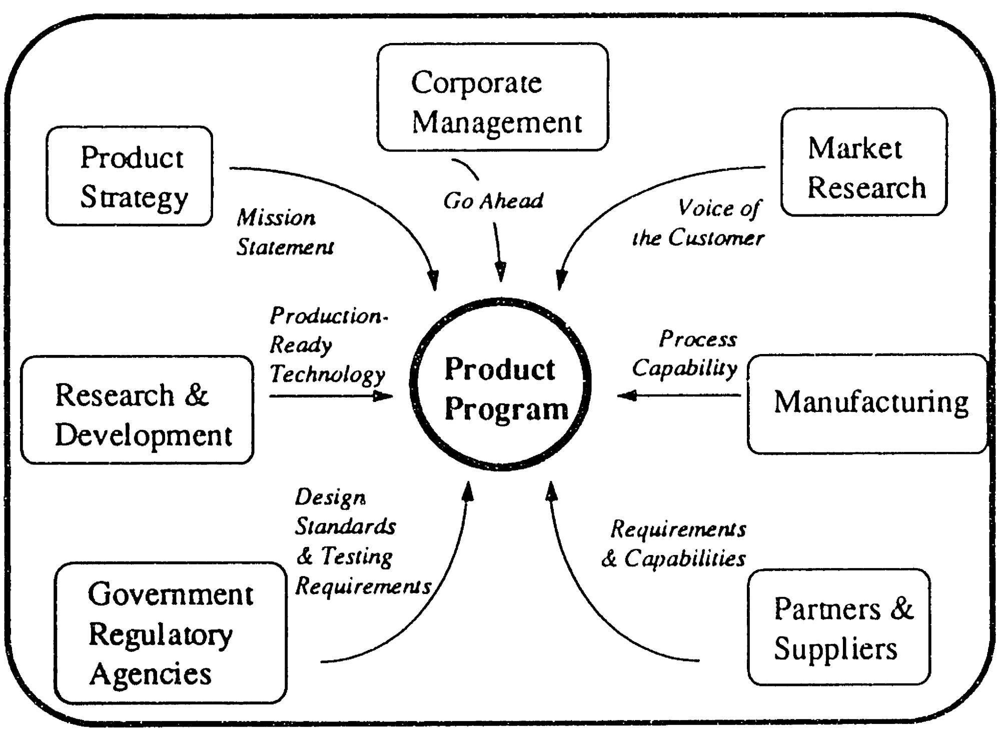

Figure 2.5: Organizational Influences of the Product Program

II-C) Systems Engineering Approach

"The whole is more than the sum of its parts.
The part is more than a fraction of the whole."

This quote, from Aristotle's *The Composition Law*, is at the core of systems thinking and systems engineering. The underlying challenge of systems engineering is to draw out, in unison, the attributes of the component parts to achieve the desired function of the total system. The process involves deciding upon the goals for the overall system, selecting the component parts, and then developing the components together to achieve the system goals. In this process, careful attention must be paid to understanding the interactions among the components. The objective is to harness the benefits of the desired interactions and to minimize the interference of the undesired interactions. In undertaking this endeavor, one begins to appreciate the profound truth of the above quote.

A more succinct definition is provided in the book *Why Systems Engineering*:

> System engineering is the formal analytical and planning model (tool) for progressing from mission objectives to achievement of those objectives in an orderly controlled manner while ensuring that all parts in the total system are integrated and functional (Corrigan & Kaufman 43).

III. Underlying Philosophy

The philosophies of (A) Function Driven, (B) Top Down, and (C) System-Wide Focus were developed during the process development team activity in which we discussed what the concept development process ideally should be. Similar philosophies can be found in the research and writings on the product development process and they are key elements of systems engineering. Essentially they embody a systems engineering approach to the conceptual design process. However, these philosophies are not commonly expressed as concisely or applied in practice to the concept development process. In fact, an approach to concept development almost 180 degrees apart is more commonly practiced. In the following sections, the application of these philosophies to the concept development process will be described and contrasted with how it is commonly practiced today.

III-A) Function Driven

Presently, many companies have become very good at bringing in the voice of the customer and "gathering requirements" for the new product they are designing. They then generate product concepts, or physical elements, to satisfy those requirements. This process is depicted in Figure 3.1.

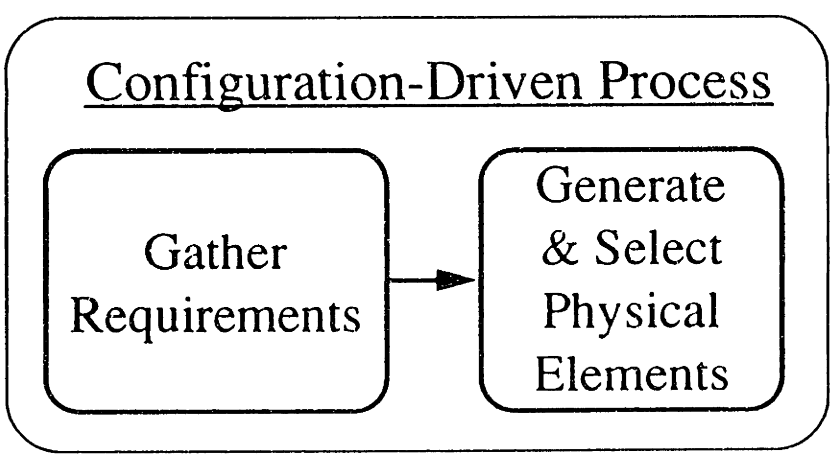

Figure 3.1: A Configuration-Driven Concept Development Process

It is almost a natural response for a design engineer to suggest physical components that satisfy specific customer needs. In a large product development organization, with several groups of engineers, competing technologies will arise and a series of product configuration proposals will be generated. The most viable configurations will usually be developed further and detailed product proposals will be presented. A decision making board, often from outside the product team, will then select the configuration to be carried forward.

The critical difference in philosophy here is that product functionality is developed before the physical design is configured. One can view the product design process as problem solving process: the problem being how to satisfy the requirements of customers and other stakeholders of the project. However, these requirements are collected in the customers and stakeholders own words, and not in the form of engineering parameters. Therefore, to begin suggesting physical elements to satisfy those requirements is to jump to the solution before really understanding the problem; a partial understanding rarely leads to more than a partial solution. The importance of developing the functionality is in its ability to describe the problem in a way design engineers can understand. And then, only after fully understanding the problem , will the team begin to generate potential solutions. The hope is that these solutions will be complete solutions that fully satisfy the requirements. This process is depicted in Figure 3.2 below.

The challenge here is to view the customers as purchasing the functionality that satisfies their requirements. This is very difficult for engineers, who, almost by their nature, think in terms of physical components and design solutions. However, this is the fundamental difference between a *market-in* and a *product-out* mentality: whereas product-out focuses on the product as the purpose of the work, market-in emphasizes customer satisfaction as the ultimate goal (Shiba, Graham, and Walden).

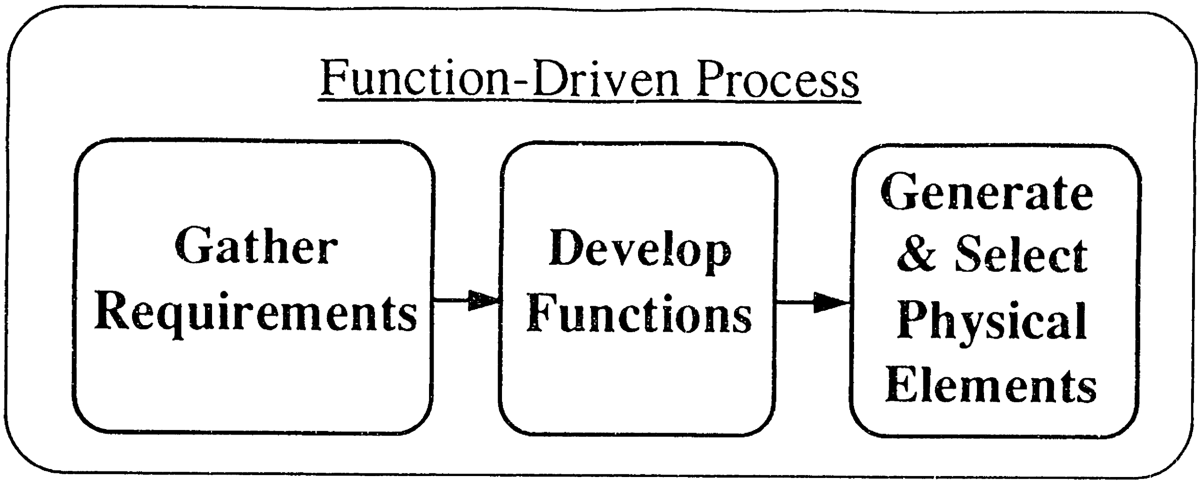

Figure 3.2: A Function-Driven Concept Development Process

This approach is also analogous to the systems engineering process. As Corrigan & Kaufman explain in *Why Systems Engineering*, "You cannot make a relevant design decision without specifying the functions that your total system must perform." (p15). They further proclaim: "System functional analysis must proceed design analysis." (p32). Therefore, systems engineering provides another strong source for a function-driven approach.

In the configuration-driven process, the products functional relationships are often developed, but only after the product concept has been selected (see figure 3.3 below). Usually, the functional relationships become more fully understood during durability and safety testing, as the development team tries to debug the product at a systems level. The failure mode analysis is often a primary driver in drawing out these functional relationships. However, they are usually developed after the fact: after many design decisions have been made.

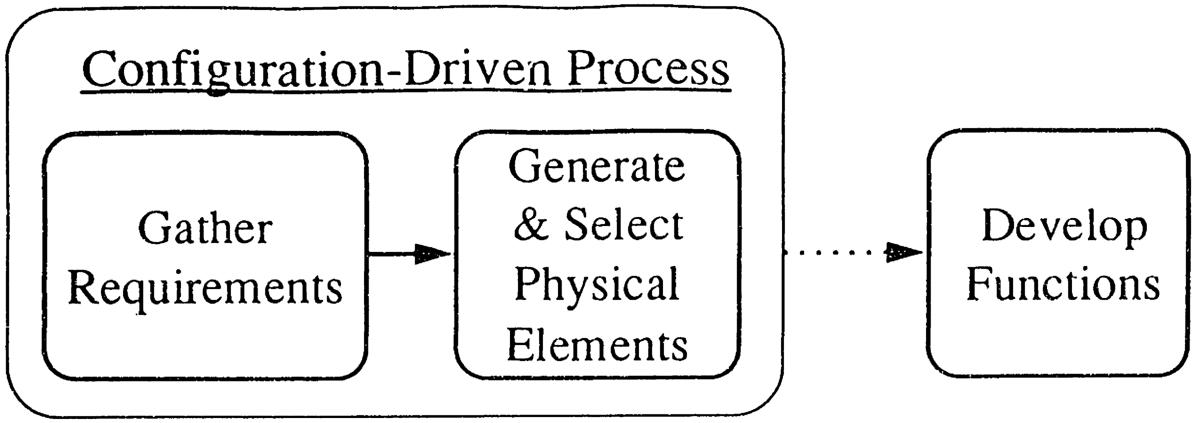

Figure 3.3: How Functionality is Handled in a Configuration-Driven Process

III-B) Top-Down

Viewing the product development process as a problem solving process helps bring us to our next underlying philosophy, *Top-Down*. Perhaps the most efficient way to solve a very complex problem is to break it down into its constituent elements, into simpler "subproblems." A top down, or hierarchical, approach to engineering design is not new: it is one of the primary tenets of systems engineering (Hall 57).

In fact this approach to systems design is well recognized. Many product development organizations are structured in a top-down manner. However, that is often precisely the problem. Organizations are typically arranged by functions in a hierarchical structure and the product architecture is then essentially forced to conform to the organizational structure (Henderson & Clark 16). The top down approach we advocate here entails deriving the hierarchy of product functionality, based on the requirements for the specific product, and *then* developing the organization around this functional hierarchy. This will be explained in more detail in section V-C.

In addition to the logic of applying a hierarchical approach to the problem solving process of product development, a top-down approach provides a much needed framework for dealing with functions. While functions are meaningful in engineering terms, they are fairly abstract and difficult to handle in a design context. Unlike physical parts and their designs, functions can't be seen, held in one's hand, or manipulated on a computer screen. In order to obtain a structure for working with these abstract entities called functions, we turn again to the field of systems engineering, which has been designing in functions (as opposed to parts) for over 50 years (Hall 11).

The Function Tree

A function tree is a tool for recording and displaying the development of the functional hierarchy. It is built in a top-down process, where each function is decomposed into it's constituent subfunctions. An example of a function tree can be seen in Figure 3.4. This function tree was created as part of a simulation exercise to design a new airplane using the concept development process described in this paper. We begin with the function 'Transport Contents,' which covers the broad mission of this product. Transport Contents is therefore considered the *top level function* and the rest of the tree will be built underneath it.[2]

The subfunctions of Transport Contents are listed one level below. The process of decomposing a function into subfunctions is fairly rigorous and requires several iterations to arrive at the right set of subfunctions. Those subfunctions, then, will directly *contribute* to the performance of the function in question while being *independent* of one another. The details of this process, including what is meant by the terms independent and contribute, will be described in section IV-E.

[2] Visually, this will resemble roots spreading out beneath a tree rather than limbs branching out above it. The convention, though, is to label these decompositional structures as trees; as in decisions trees, family trees, etc.

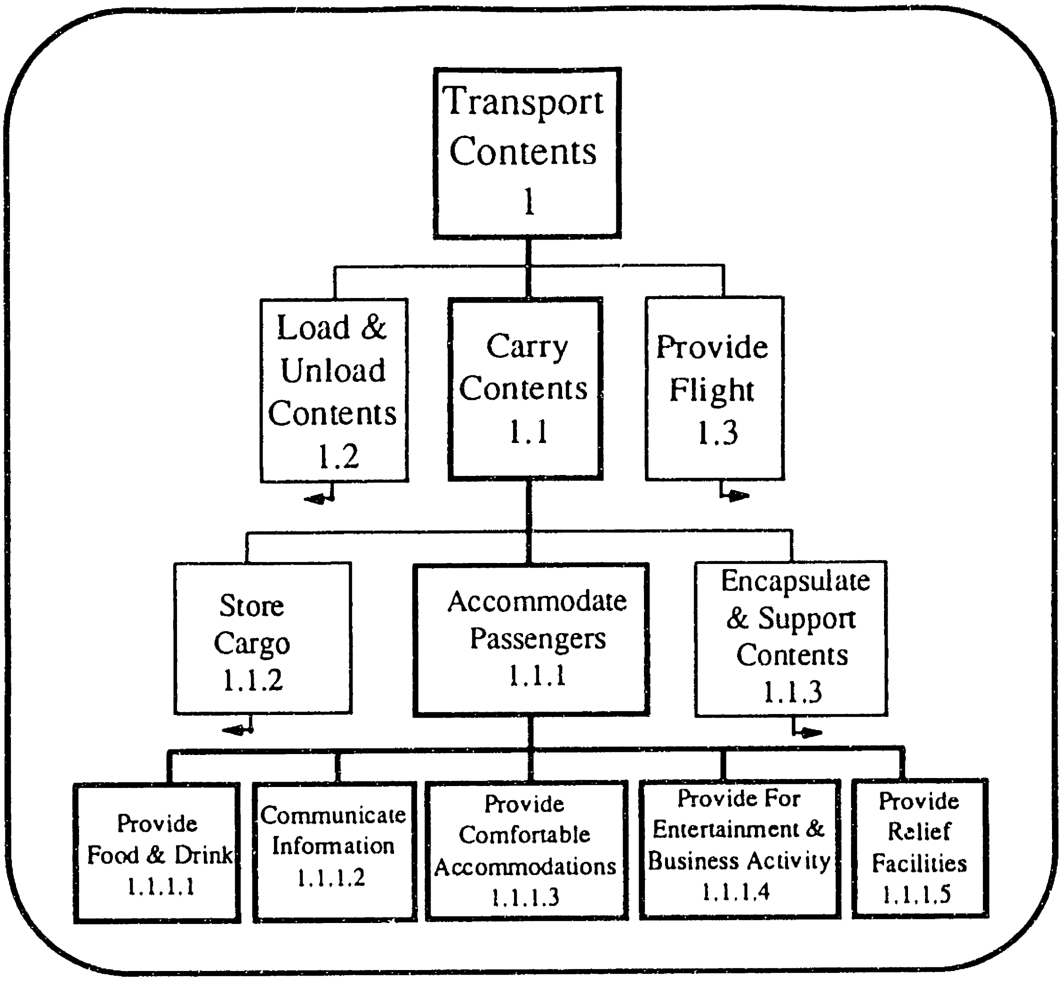

Figure 3.4: The Function Tree
A Top Down Approach

The function tree shown here represents the decomposition of only one branch. Each function will usually spawn between three and six subfunctions.[3] Therefore, the function tree will spread out very fast. For the purposes of this study, we were more interested in understanding the behavior of the process through increasing levels of detail and therefore

[3] The terms function and subfunction are only used relative to one another. Each block on the function tree represents a function; however, that function is also referred to as a subfunction with respect to the function it was derived from. For example, while Carry Contents [F 1.1] can be described as a *function* having the subfunctions Accommodate Passengers [F 1.1.1], Store Cargo [F 1.1.2] and Encapsulate & Support Contents [F 1.1.3]; Carry Contents can also be described as a *subfunction* of the top level function Transport Contents [F 1].

focused on developing the depth of the tree rather than the breadth. As a result, only one function is carried down at each level. Those functions that are not carried down have an arrow protruding from their base to indicate the presence of further subfunctions not shown here.

The function tree is powerful tool for visually depicting the hierarchical structure of the functional relationships. It is just one means to handle functions, enabling one to "design in functions," rather than in physical elements or parts. Other tools of systems engineering, such as N^2 Diagrams and functional schematics, are also used to further develop and depict functional relationships and functional interactions. (These will be described in section IV-E.) These tools provide an analytical framework and formal guidelines for handling functions which is missing from the early stages of product design.

An example depicting common practice will demonstrate the pitfalls associated with not using an analytical framework for developing functionality. Figure 3.5 shows some of the functions a committee designated as the top level functions for a new aircraft. These functions were not developed in a top down manner, but were based on the systems found in existing airplanes and the functional groups which designed those systems.

The first drawback of this approach is that these functions are rooted in the existing functional groups of the organizational structure. Therefore, the organization winds up driving the design process and thus the ultimate concept selection, rather than the process driving the structure of organization. In anchoring itself to the organization of previous designs, it opposes a forward-looking approach to product development. Finally, and most importantly, this approach does not apply formal guidelines for dealing with functions.

The lack of formal guidelines lead to several potential problems, including redundant functions and functions of different levels. For example, 'Perform Flight Management' and 'Manage and Control Aircraft' are two functions that seem to have a lot of overlap. It is very easy to include unintentional redundancy when functions are based on systems from previous products. Furthermore, by not distinguishing between the levels developed on the function tree, this approach raises serious difficulties in handling functions. This is illustrated in the following example.

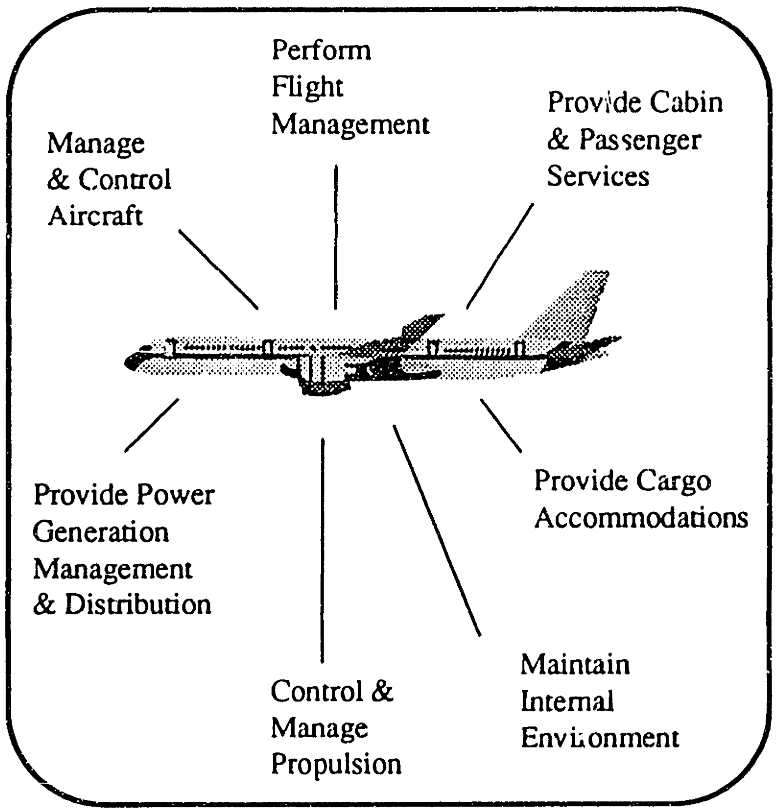

Figure 3.5: Functional Development Based on Existing Product/Organization

In Figure 3.5, 'Provide Cabin & Passenger Services' is listed at the same level as 'Maintain Internal Environment.' However, Maintain Internal Environment is really a subfunction of Provide Cabin & Passenger Services: If there were no function to provide a cabin there would be no internal environment to maintain. Going back to the problem solving analogy, a subfunction can be thought of as a subproblem of the higher level problem. However, if no distinction is made between a subfunction and a function, the scope of the problem to be solved will not be defined. The result will be overlapping solutions that are not necessarily compatible. Unless there is an orderly means to deconstruct the design problem into simpler pieces or subproblems, there will be no clear and easy way to bring the solutions together to solve the higher level problem.

Moreover, by not rigorously distinguishing between a function and its subfunctions, the task of defining functional relationships and interaction becomes exceedingly complex. The challenge in designing in functions as compared to parts, as explained above, is in understanding how they relate and interact with one another, since this is not as readily apparent. A very helpful tool for describing these is the functional schematic (see Figure 3.6), which depicts the interactions of subfunctions in the performance of a function.

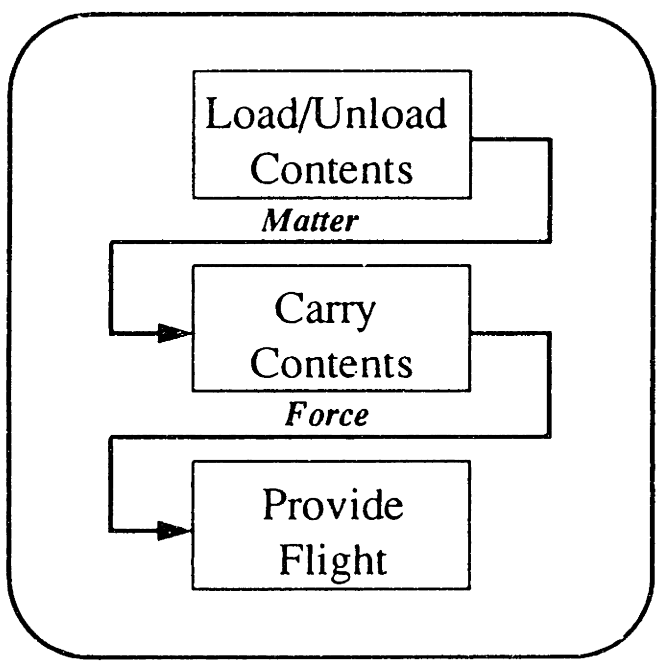

Figure 3.6: Functional Schematic for 'Transport Continents'

Drawing an accurate schematic diagram requires a separation of function and subfunction. Without that clear distinction, functional interactions become extremely difficult to sort out. As a result, designing in functions becomes too confusing, leading design engineers to fall back on the process they are more familiar with, a configuration-driven approach.

The function tree facilitates a function-driven approach, not only by clearly depicting the hierarchical relationship of functions, but also by suggesting an orderly process for deconstructing the design problem. An additional benefit of the function tree is that in presenting the big picture view, it enables visual inspection to prevent redundancy and to promote completeness. Finally, the function tree provides the design team with an overall system focus, which is the subject of the next section.

III-C) System-Wide Focus

Even with the systems engineering tools described above, designing in functions is still a very difficult activity. Dr. Don Clausing, who advocates using the function tree to guide the design in *Total Quality Development*, was able to offer some further guidance. He explained, "the levels of a function tree are not unique: different teams will come up with different trees." We found this to be the case during our process simulation. While it was often difficult to decompose a function into its constituent subfunctions, it was even more difficult to develop the functions for the next sub-level. At the second level down, the decisions we were making seemed too arbitrary. Therefore, something was needed to ground the tree in order to make it more meaningful.

We learned that a physical element tree, or a "hardware tree" as Clausing calls it, developed in conjunction with a function tree will ground the process. A physical element tree is simply the collection of physical components that best satisfy the corresponding function, given the customer requirements and design constraints. Figure 3.7 depicts a physical element tree developed in conjunction with the function tree shown in Figure 3.4.

Now that we had the pieces, namely the function tree and the physical element tree, the challenge was to figure out how to put them together. We went about this in two ways: process simulation and process research. In the simulation exercises we experimented with different methods for transforming functions into physical elements. The research activity brought in design philosophy as well as tools and templates for carrying out the process. However the one thing that helped us progress the most was the following guideline for alternating between the functional realm and the physical realm: "Subfunctions of a function can only be determined by the consideration of the hardware concept that performs that function" (Sontow & Clausing 30).

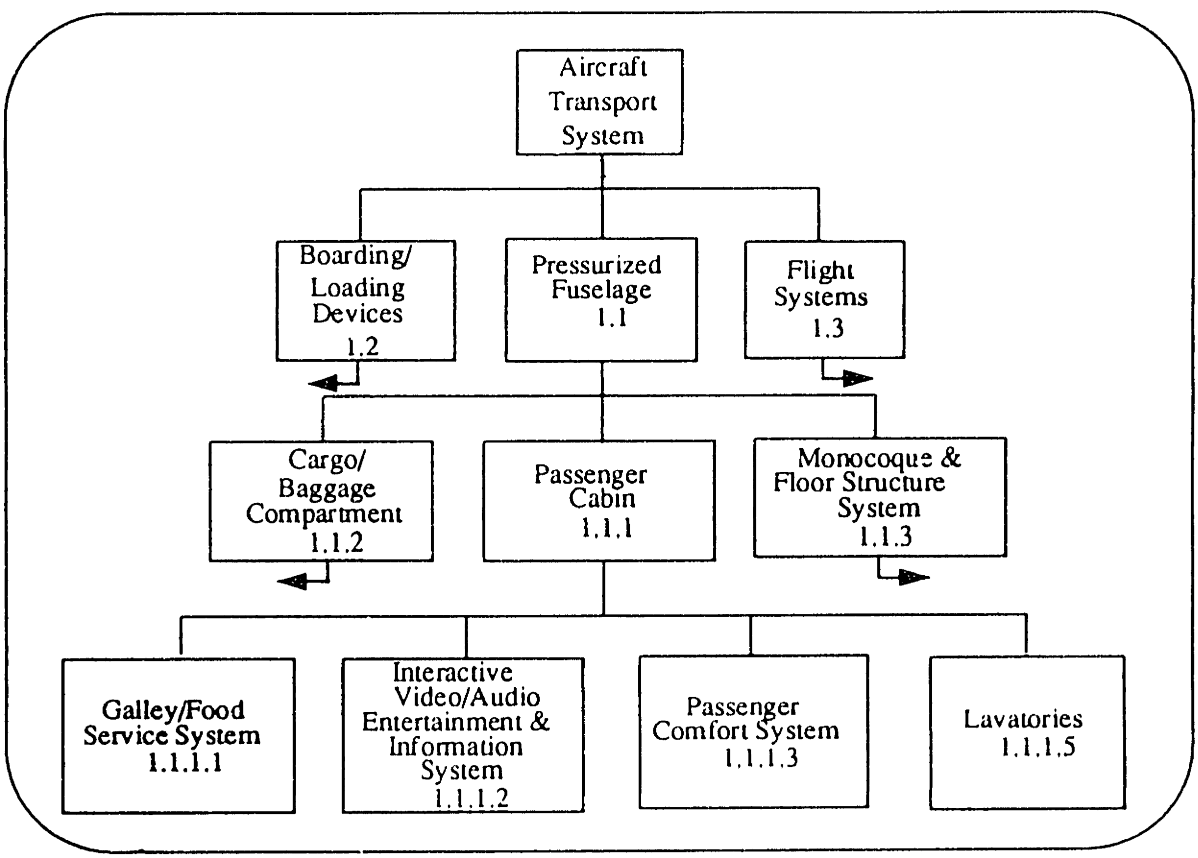

Figure 3.7: Physical Element Tree

The essence of this statement is that a function only becomes grounded when associated with a physical element to carry it out. Without a *notion* of the physical element to be used to satisfy the function, there is no objective means to identify the best group of subfunctions. Moreover, this should be the case: the purpose of a function-driven process is not to construct a function tree, but to keep the design focused on the problem to be solved.

Furthermore, in selecting the best physical element concept to perform a function, one must have a sense of the subsystems and components in order to distinguish between the different concepts. Therefore, a physical element is not really defined unless we have a notion of the subsystems and components that make it up. For example, a Pressurized Fuselage [PE 1.1] can take on many forms; however, it's image becomes much more concrete given the following subsystems: a Passenger Cabin [PE 1.1.1], a Cargo/Baggage Compartment [PE 1.1.2], and a Monocoque & Floor Structure System [PE 1.1.3].

This philosophy led us to develop the process, depicted in Figure 3.8, for cycling between functions and physical elements. The diagram shows the process of decomposing a function into subfunctions, as described earlier; however, it also shows a dotted line emanating from the top level physical element and pointing to the top level function. This line is meant to signify that a *notion* of the physical solution for the function is necessary to develop the subfunctions. In the example presented above, we were able to decompose Transport Contents [F 1] into Carry Contents [F 1.1], Load & Unload Contents [F 1.2], and Provide Flight [F 1.3] based on the notion that we were designing an airplane. Obviously, if we were designing an ocean liner we would need to provide float, not flight.

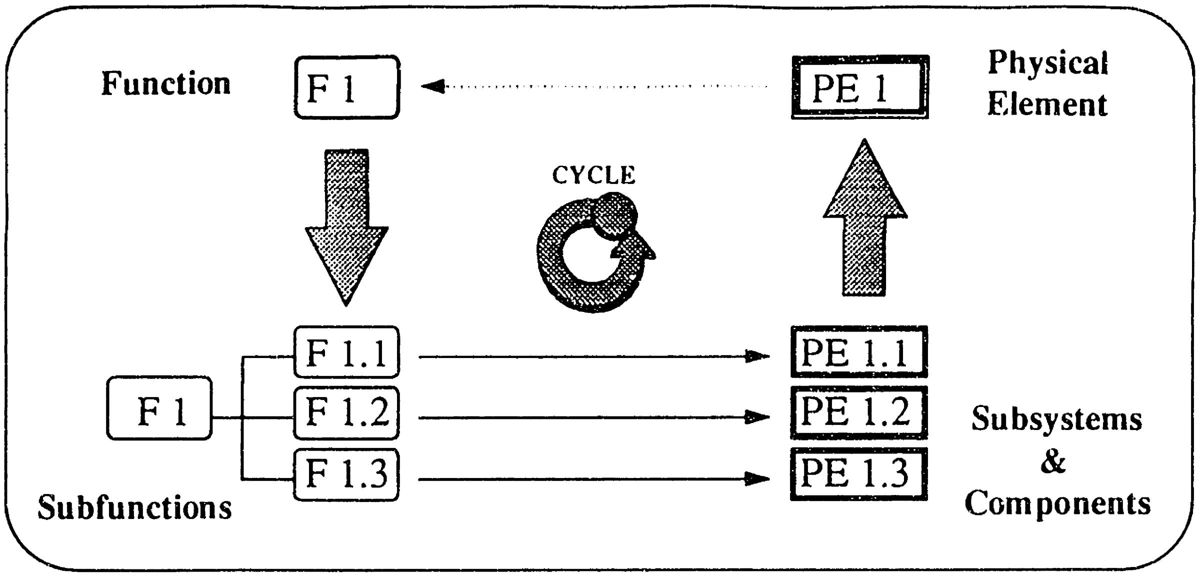

Figure 3.8: Process for Cycling Between Functions & Physical Elements

After we develop the subfunctions, we proceed from the functional realm to the physical realm in order to develop potential subsystems and components that provide the desired functionality. Figure 3.8 depicts this process in a simplified form for the sake of clarity. However, there are three important aspects of this process that are not depicted: the mapping of more than one subfunction to one subsystem, the generation of several alternative subsystems, and the cycling down of this process to the lower levels.

Figure 3.8 only shows a one to one mapping of subfunction to subsystem. In reality you may develop a subsystem that can satisfy more than one subfunction, as depicted in Figure 3.9. Here PE 1.2 is an elegant solution that satisfies two functions with one stone, so to speak.[4]

[4] However, the case of one function mapping to two physical elements would indicate that the subsystems and components listed are at a lower level. In that case one would have to suggest a higher level physical solution.

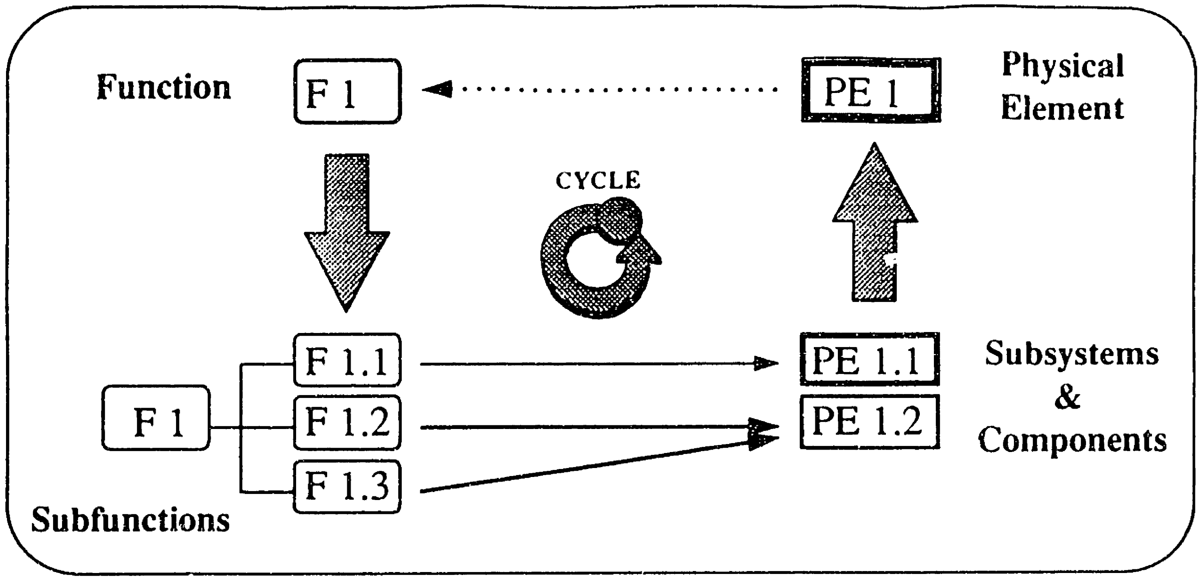

Figure 3.9: Opportunity for One Physical Element to Satisfy Two Functions

An example of a two-to-one mapping, from the function and physical element trees presented earlier, is how the functions Communicate Information [F 1.1.1.2] and Provide for Entertainment & Business Activity [F 1.1.1.4] are both satisfied by an Interactive Video/Audio Entertainment & Information System [PE 1.1.1.2] .

The process depicted in Figure 3.8 has been additionally simplified in that it only develops one complete set of subsystems and components to satisfy the functionality. The hope is that in practice the design team will be able to generate several different types of subsystems and components to satisfy the required functionality. This one step requires the most creativity and far reaching thought in order to discover new and better solutions. Then, the design team can mix and match the subsystems and components to generate several feasible sets, or system-level physical element concepts. The concept which as a system best satisfies the requirements and subfunctions of the higher level function may then be selected from among the feasible alternatives. This selection process, accomplished via the Pugh concept selection matrix (which will be described in Section IV-H), results in defining the top level Physical Element ([PE 1] in Figure 3.8).

In combining the subsystems and components into feasible concepts a *system-wide focus* is emphasized. Rather than solving one subfunction independent of the other subfunctions it interacts with, this approach generates physical systems that *solve all the subfunctions simultaneously*. Not only will this serve to minimize compatibility problems lower down in the design process, but it will also enable optimizing the product design by selecting concepts based on their performance as a system. This, of course, is key component of systems engineering, as Hall states in *Metasystems Methodology*: "Systems methodology is really the same as for other applied sciences, yet its outlook is different in its emphasis on the total system-environment interaction."(5). Thus, one of the primary benefits of applying a systems engineering approach to concept development is the overriding focus on developing concepts based on how they function as a whole.

One final aspect of this process to be addressed is how this cycle extends into greater and greater detail, into lower levels of the function and physical element trees. At the end of the cycle described above (and depicted in Figure 3.8) the top level function and physical element were defined, and could then be incorporated into their respective tree. Then, the cycle continues with each subfunction being decomposed into its respective subfunctions. Figure 3.10 depicts this process for F 1.1. In this way, the process continues to explore the subfunctions and subsystems at the next level of detail in order to develop the physical element which best provides the desired functionality, or the physical solution to the functional problem.

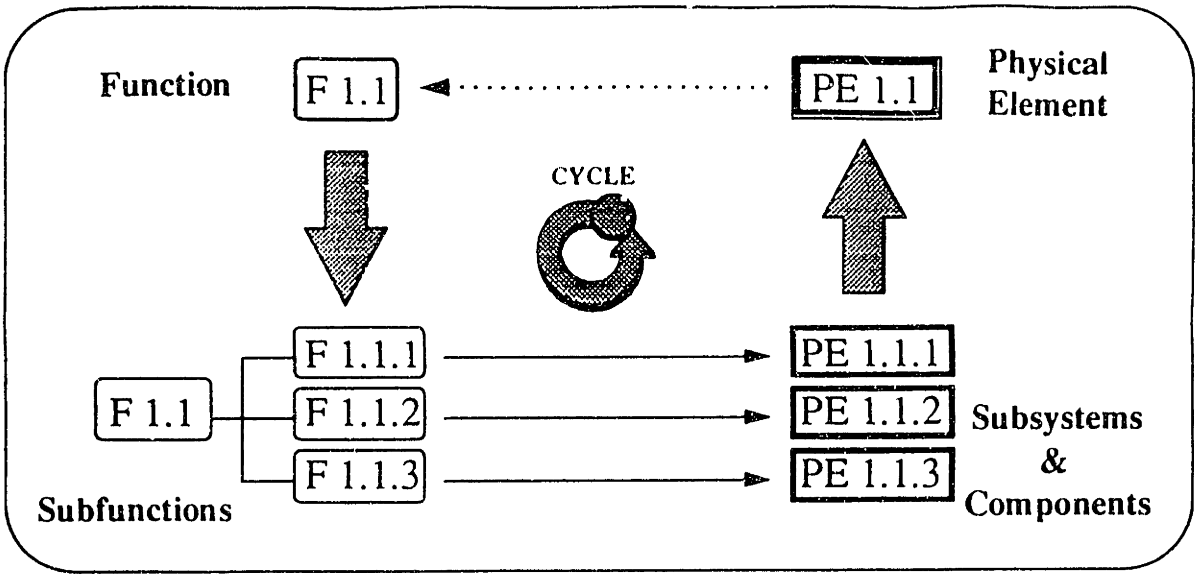

Figure 3.10: Cycle Depicted for the Next Level Down

After developing this process, it was reassuring to find research that supports the approach we developed through our simulation activity. The following is a quote from Nam P. Suh's *The Principles of Design*:

> Design involves a continuous interplay between *what we want to achieve and how we want to achieve it.* For example, on a grander scale, we may say "what we want to achieve" is to go to the Moon, where as "how" is the physical embodiment in the form of rockets and space capsules. On a smaller scale, "what we want to achieve" may be the minute amount of moisture in plastic pellets and "how" may be the special titration instrument...These examples show that the *objective* of the design is always stated in the *functional domain,* whereas the *physical solution* is always generated in the *physical domain.* The design procedure involves interlinking these two domains at every hierarchical level of the design process. These two domains are inherently independent of each other. What relates these two domains is the design (25-26).

IV. Steps of the Process

IV-A) Overview

In this chapter, the steps of the process will be described and illustrated with examples developed during the simulation activity. The steps, and the process flow, are depicted in Figure 4.1.

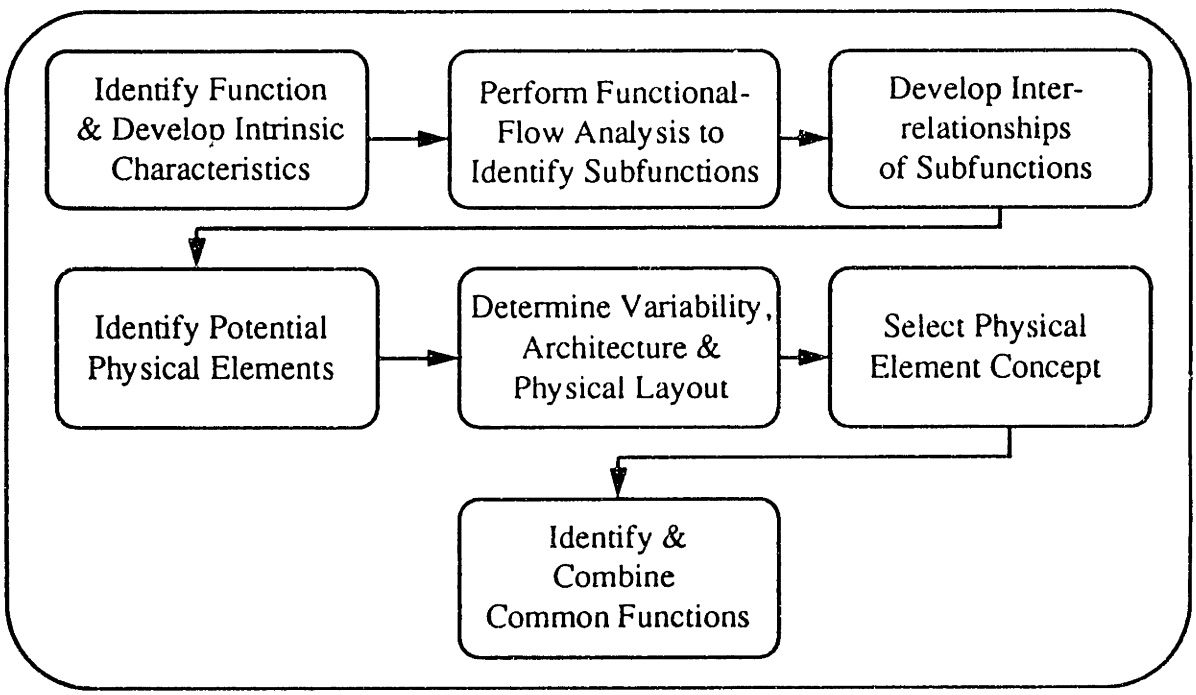

Figure 4.1: Steps of the Concept Development Process

Before the process steps are described, a few points need to be emphasized. The process presented here is a guideline for generating and recording the necessary information to carry out the concept development process: it not a recipe to be followed without question. In fact, the process should be applied in the spirit that it was developed, by continually seeking ways to improve it and questioning its limitations. It should then be clear that in no way is this meant to serve as a replacement for creative, independent thought. It is, however, designed to serve as a means for channeling individual and team creativity towards more productive ends. Never-the-less, many steps, and perhaps the entire process, will require several iterations before the results are deemed to be "correct." And that determination is largely based on the judgment of the process participants, who will have to develop the sense of when a concept has been developed solidly enough to proceed to more detailed definition. Several rational checks and balances are built into the process to assist in developing that sense, but it still involves a learning process. Finally, the goal is not to complete these steps: it is to generate and synthesize the data required to make the right design decisions and to record them for future reference.

IV-B) Identify Function

In keeping with a function-driven approach, the first step of the process is to identify the function that will drive the cycle depicted earlier in Figure 3.8. The top-level function is developed based on the mission statement for the product and given a notion of the likely physical element. The product mission statement, which is an output of corporate strategy, forms the basis of the program.[5] Usually, the general notion of the physical element will be predetermined. For example, in the Boeing Commercial Aircraft Group, the top-level physical element will be an aircraft, unless otherwise explicitly specified.

Given the physical notion, and the product mission statement, the team identifies the function in broad enough terms to encompass the overall mission of the product in a very few words. A function is ideally stated in a two word verb-noun phrase. This is for reasons of consistency, clarity, and especially for ease of communication. If the functions were described in complex sentences, as they often are initially conceptualized, it would be nearly impossible to employ graphical tools such as a function tree, an N^2 Diagram, or a functional schematic with any clarity. As verbal detail increases, these tools lose their ability to be used in team settings (as one is required peer at them with a microscope held millimeters above the surface). Therefore this concise format enables us to cover a lot of

[5] For an aircraft product, the mission statement would most likely include target payload capacity, range, speed, operating efficiency, product costs, and program investment.

breadth at each level of depth, and thereby ensures the system-wide focus so valuable to this process.

In the case of our simulation activity, the top-level function was identified as "Transport Contents." In practice, the top-level function is the only function that will be formally developed in this step. Since subfunctions will be developed in a later step of this cycle, the lower level functions are simply taken from the subfunctions developed in the previous round. Furthermore, the physical element notion grounding that function will also be based on the subsystems suggested in the previous round.

IV-C) Develop Intrinsic Characteristics

The Intrinsic Characteristics are the requirements that define the product: without satisfying them the product will not fulfill its mission. They are also useful in helping to frame the problem we are trying to solve. In doing so, the intrinsic characteristics help focus effort in developing the subfunctions and serve as selection criteria for choosing the best system concepts.

The term intrinsic characteristics was chosen, in contrast to customer requirements, to convey the notion that a product is required to meet specifications originating from several sources in addition to the end customer, including: 1) Government Regulatory Agencies; 2) Corporate Product Strategy Office; 3) Partners/Suppliers; and 4) Manufacturing Facilities. The ultimate goal of the product development team is then to satisfy the customer as well as possible while meeting the constraints imposed by these other stakeholders.

Figure 4.2 provides a view of how developing intrinsic characteristics fits into the larger product development process. It is in a sense the bridge between the market research function, which serves to collect the requirements, and the concept development activity.

In this step the requirements are brought together and organized to create the framework within which the concept development team must operate. This can be thought of as defining the solution space for the design process. Product concepts that fall within that solution space will satisfy the customers and meet the constraints. Furthermore, in developing the intrinsic characteristics, the requirements are prioritized to lay out the path for best satisfying the customer.

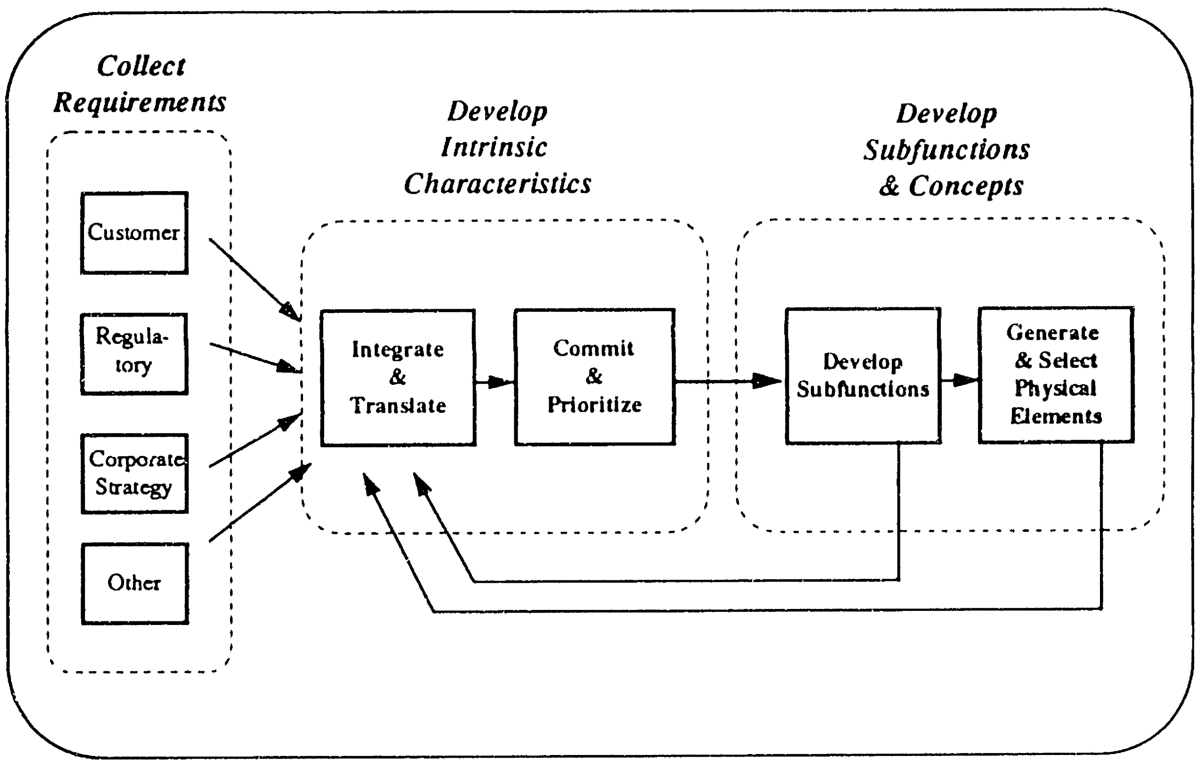

Figure 4.2: Develop Intrinsic Requirements

In addition to the requirements gathered from the customers and the stakeholders mentioned above, requirements will also filter down from previous cycles of the concept development process. The functionality and the physical concepts chosen in previous rounds will feed requirements to the lower levels. For instance, the intrinsic requirements for a function will filter down to its subfunctions. Likewise, the system requirements will spur specific subsystem requirements. This process is depicted by the left-pointing arrows in Figure 4.2. Again, it is important to point out that these requirements are coming down from the previous round (and are not feedback loops).[6] Appendix A contains examples of how intrinsic requirements filter down for functions at different levels of the tree. Those examples were taken from our simulation activity and are disguised to prevent disclosure of any proprietary information.

The product development team must integrate all the requirements, grouping similar requirements into categories. During this process, conflicting requirements will be analyzed. The team will first try to clarify the requirements, and in doing so hopefully resolve the conflict. However, if the requirements are truly mutually exclusive, then a decision will need to be made as to which of the conflicting requirements the team will commit to.

Once the requirements have been integrated they need to be translated into design terms. In the Quality Function Deployment (QFD) process this step is called translating the voice of the customer into product expectations or engineering characteristics. The terminology we chose for the translation process *develop functionality*, meaning that we translate the intrinsic characteristics into functional requirements for the design. The team strives to develop measurable performance parameters; however they must also capture the subjective characteristics important in performing the given function.

---

[6] Some requirements, such as ownership and operating costs, filter down in detail through the levels. Thus, overall costs are allocated to a specific function and are then divided up among its subfunctions. Lead time, or time-to-market, is another example of an overall requirement that filters down as a constraint to lower levels of intrinsic characteristics. The individual time-to-market requirements are based on the lead time for the entire program and are then adjusted according to how the component in question fits into critical path timing. Other characteristics will increase in level of detail as they are filtered down. For example, while the top level intrinsic characteristics will state the requirement to comply with regulatory standards, lower level intrinsic characteristics will call out the specific standards. Finally, there will be certain characteristics not appearing at higher levels that are brought in at lower levels. These will usually stem from the selection of a particular subsystem or component. For example, given the notion of a Faucet/Sink System [PE 1.1.1.5.5] to satisfy the Provide Hand Cleaning Facility function [F 1.1.1.5.5], the requirement for water meeting the standard of potability is introduced. Prior to this requirement for water on the airplane, there was no need for a water purity standard.

There is a formal process for committing to the intrinsic characteristics. This provides a check point to ensure that team believes the requirements are attainable and that the list doesn't include any mutually exclusive requirements. Once they have been committed to, the requirements need to be prioritized to distinguish the most essential characteristics from those of lesser importance. This will help the team focus on providing the most important requirements first. In addition, the characteristics could be categorized into groups based on whether they lead to customer satisfaction or to avoidance of dissatisfaction.[7]

IV-D) Perform Functional Flow Analysis To Identify Subfunctions

The functional flow analysis is a key step in decomposing a function into its constituent subfunctions. It is accomplished by deconstructing the function into a series of steps required to perform that function. Doing so will often require a notion of the physical element expected to solve the function. An example of the functional flow for Transport Contents, given the notion of an aircraft transport system solution, is shown in Figure 4.3.

[7]Kano analysis provides a methodology for categorizing customer requirements into the following groups: attractive, one-dimensional, must-be, indifferent, and reverse quality elements. This process quantifies the way a requirement needs to be met in order to provide satisfaction or to avoid dissatisfaction, as the case may be. For more information, see Burchill's *Concept Engineering* text listed in the Reference section.

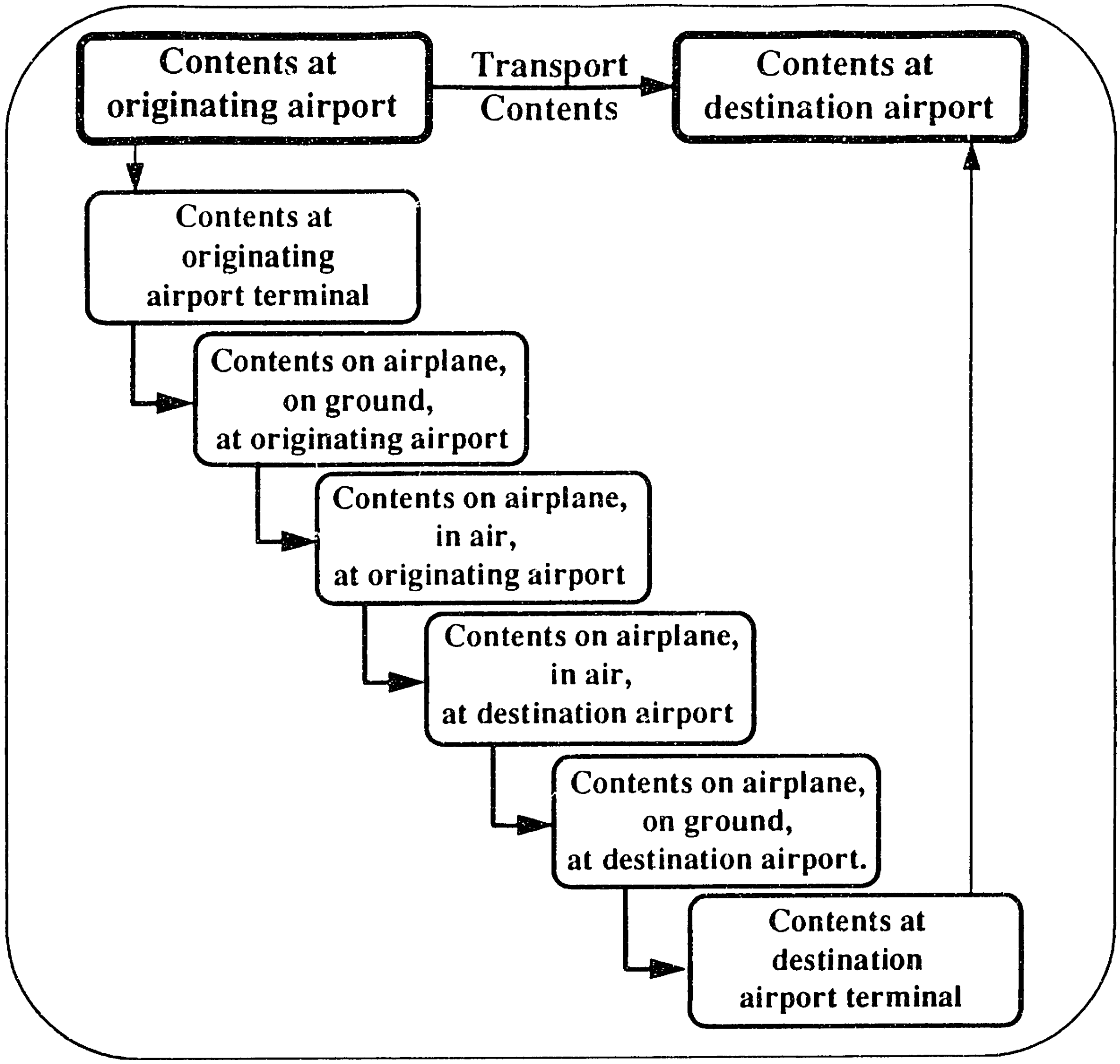

Figure 4.3: Functional Flow Analysis for Transport Contents

The importance of the functional flow analysis is that it provides the framework for decomposing a function into its component subfunctions. After deconstructing the function into a series of steps that constitute the functional flow, the subfunctions are derived by solving the problem of how to get from one step to the next. This is achieved by arranging the functional flow steps into an Initial State-Subsequent State analysis, shown in figure 4.4.

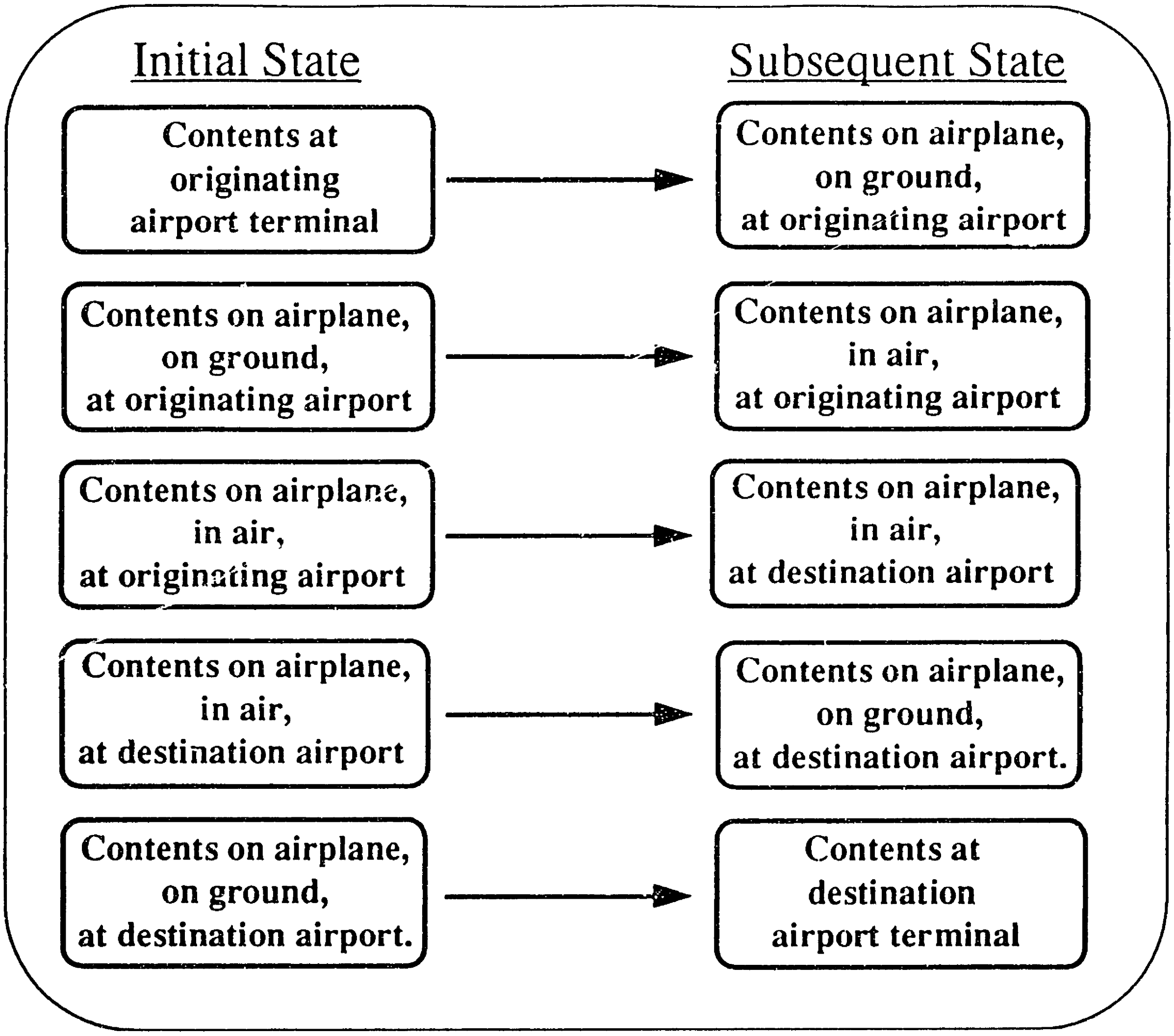

Figure 4.4: Initial State - Subsequent State Analysis for Transport Contents

In order to complete the Initial State - Subsequent State analysis, the question of how to get from the initial state to the subsequent state is posed in the following way: "Is there a transfer or transformation of matter, force, energy, or information?" Figure 4.5 depicts this question graphically and Figure 4.6 shows the answers to this question for the Initial State - Subsequent State analysis. (Note: in Figure 4.6 the term *force* is used to signify both force and energy, which are often transferred/transformed together.)

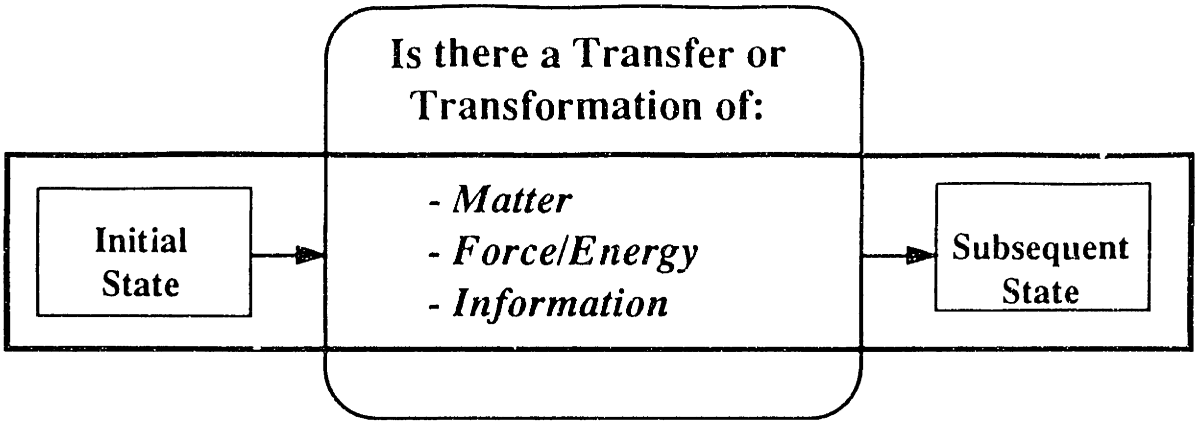

Figure 4.5: "What Brings About The Change In State?"

This question was something we developed to help guide us in the development of subfunctions. Without asking this question, identifying the subfunctions that bring about the change of state often felt like grasping for the ether.

In our process development simulation, we also added the category of "Environmental Conditions" as a convenient grouping of pressure, humidity, temperature, etc. While "environmental conditions" is a combination of certain matter and force/energy transfers and transformations, and not a truly unique category, it helped simplify the process of identifying subfunctions and describing the functional interactions. While it simplified our process it may not make sense for other products. Each design team will have to decide which categories make the most sense for the product they are developing. However, we believe that matter, force/energy, and information comprise the minimal set of categories. [8]

[8] The categories are similar to those suggested in Ulrich & Eppinger: material, energy, and signal flows (82).

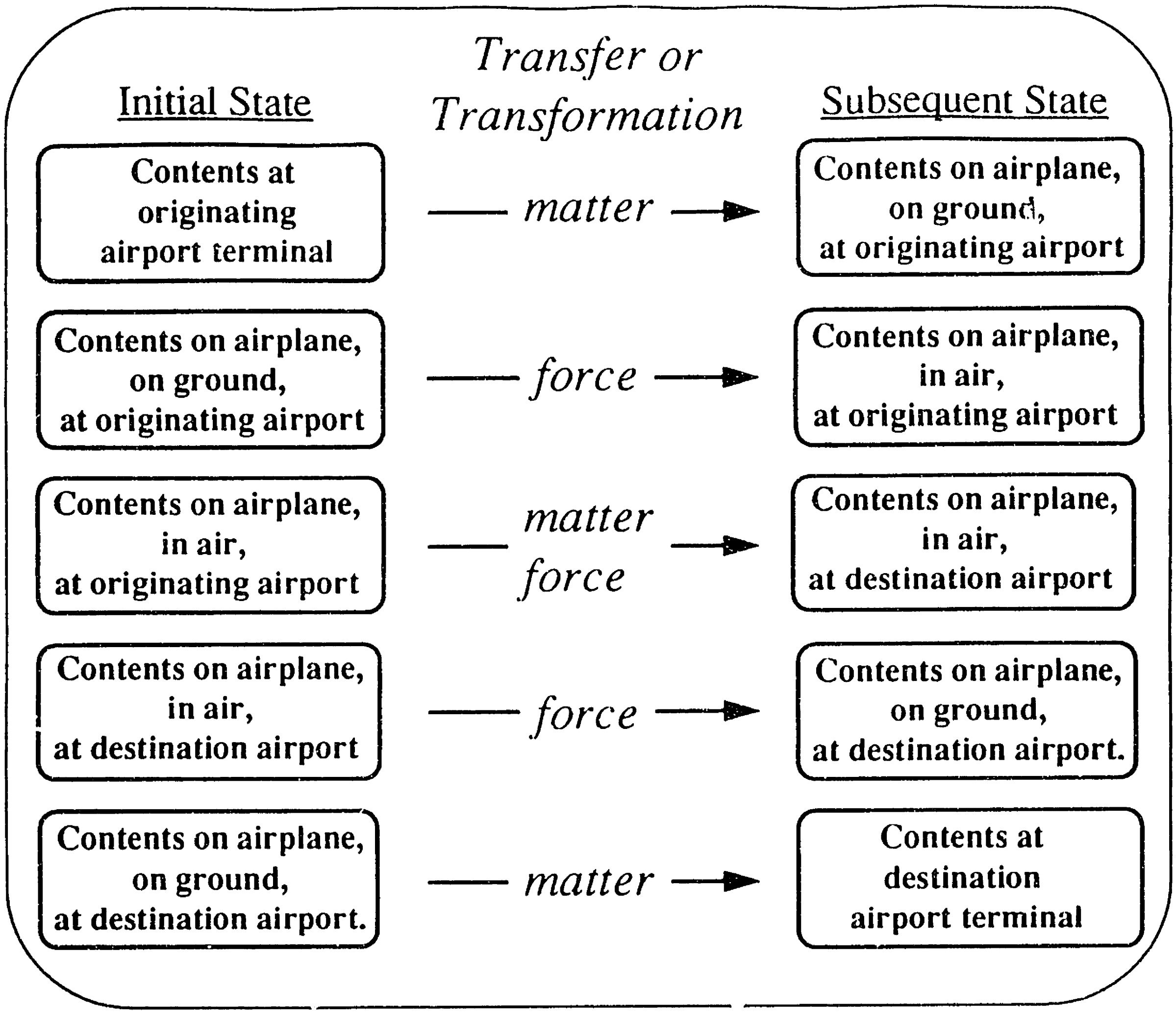

Figure 4.6: Type of Transfer or Transformation of Subfunction

The next task is to define the subfunctions that bring about the transfer and transformations identified above. Figure 4.7 displays the subfunctions identified to bring about each change in state. For example, we describe the matter transfer from the airport terminal to the airplane as "Load Contents" and we describe the force/energy transfer from airplane on the ground to airplane in the air as "Lift Airplane." As described earlier, the convention of a verb-noun phrase is used to identify a function or a subfunction. The table shown in Figure 4.7 served as a tool for writing down the steps of the functional flow and then for identifying the subfunctions that bring about the transformation from initial to subsequent state.

Initial State	Subfunction	Subsequent State
Contents at originating **airport terminal**	Load Contents	Contents **on airplane**, on ground, at originating airport
Contents on airplane, **on ground**, at originating airport	Lift Airplane	Contents on airplane, **in air**, at originating airport
Contents on airplane, in air, at **originating airport**	Maintain Flight Carry Contents	Contents on airplane, in air, at **destination airport**
Contents on airplane, **in air**, at destination airport	Land Airplane	Contents on airplane, **on ground**, at destination airport.
Contents **on airplane**, on ground at destination airport.	Unload Contents	Contents at destination **airport terminal**

Figure 4.7: Subfunction Identification for Transport Contents

While this process may appear exceedingly simple, and the results far from earth-shattering, two important comments must be made. First, this approach is the only clear approach to decompose a function into subfunctions with any degree of structure and repeatability. Most other approaches rely on the knowledge-base of an experienced designer, which is fine as long as yesterday's product architecture will satisfy tomorrow's customer. Furthermore, it is an approach that inspired the confidence of the seasoned design engineers and systems engineers who participated in the process simulation. Surprisingly, not only is it uncommon, but it is virtually unknown. The process to break down a function into a functional flow and then to develop subfunctions to accomplish the change of state was presented in a working paper by Sontow & Clausing; according to Don Clausing, it has not yet been elsewhere applied. In fact, prior to receiving the Sontow-Clausing paper, this step was the missing link in our function-driven process. As mentioned earlier, the question in Figure 4.5 was an improvement suggestion made during the process simulation activity to further assist in identifying subfunctions. In sum, this process is fairly unique and has significantly contributed to viability of a function-driven, top-down approach.

The second comment, regarding the lack of earth-shattering results, is *there is beauty in simplicity*. The example that is presented in this chapter is the result of three or four iterations through the process. Initially, the results were more confusing and there were questions as to the level of certain functions. However, after several iterations, with logical and level checks applied after each,[9] the list of subfunctions became increasingly straight-forward. Furthermore, the last iteration produced the same results as the second to last, indicating that we had obtained a fairly thorough understanding of the problem and any further simplification would sacrifice completeness. While it is difficult to convince others, the best solutions are often those that appear to be the simplest.

That said, there is one aspect of the Initial State-Subsequent State analysis that we did have difficulty with. This was in deciding what to track through during the functional flow analysis, which was not always so obvious. Initially we tracked the flow of the contents, since our prime objective was to transport them. Following the flow of the contents allowed us to develop the subfunctions for loading and unloading contents, since the contents needed to be transferred from the originating terminal to the airplane and then from the airplane to the destination terminal. However, we knew there was more to an airplane than loading and unloading contents. Therefore, we realized we also had to follow the flow of the airplane: both from the originating airport to the destination airport and from the ground to the air and back to the ground again. We also realized that we have to pay attention to those aspects of the functional flow that don't change over time. It is from those elements that the support function emerges: it is from "contents on

---

[9] A logical check simply involves rationally scrutinizing the results. When conducted in a team setting, this check uncovers gaps in thinking or flaws in reasoning. A level check will be described in more detail in the next section. It serves the purpose of sorting out subfunctions from functions, or checking to see if all the functions listed are at the same level.

airplane at originating airport" to "contents on airplane at destination airport" that we arrive at "Carry Contents." However, it was only because we were trying to arrive at Carry Contents that we chose to phrase the initial and subsequent states the way we did. Thus it is important to have an expectation of the types of subfunctions to be developed and to be creative in selecting the initial and subsequent state. If the team has gone too far in creating subfunctions, they will be able to catch their mistakes in the logical and level checks applied at the end of every iteration. Therefore, the best set of subfunctions don't automatically fall out of this process, but they can be drawn out through persistent group effort.

IV-E) Develop Inter-relationships of Subfunctions

Once we have deconstructed a function into its component subfunctions, the next challenge is to determine how those subfunctions interact to perform the higher level function. We use two tools of systems engineering to help us in this step of the process. The first is the N^2 Diagram, which enables a very rigorous analysis of all the inter-relationships between the subfunctions. The second is a functional schematic which we use to graphically present the most important interactions in a very readable format. In going through this step, we will develop a sufficient understanding of the functional relationships so that we may propose physical elements which satisfy the functional requirements of the complete system.

In this step, the N^2 Diagram is used to achieve three ends: 1) to identify interactions between subfunctions, 2) to check the level of all the subfunctions, and 3) to group closely related subfunctions. After achieving these ends, the team will have developed a fairly solid understanding of the inter-relationships of all the subfunctions and will be able to depict this in a functional schematic

Functional Interactions

The N^2 Diagram is constructed by laying out all the subfunctions along the diagonal of a square matrix. Then for each subfunction, the outputs are listed along that row and the inputs are listed along that column. An N^2 Diagram for the example we have been using is presented in Figure 4.7. The process we used to identify the interactions was to begin at the top right and work our way down the diagonal. For example, we would ask the following questions for each pair of subfunctions:

1) Are there any outputs from *Function A* to *Function B*?

2) Are there any inputs from *Function B* to *Function A*?

The outputs would then be listed in the cell where *row A* joins *column B* and the inputs would be listed in the cell where *column A* meets *row B*. The same categories used to describe the change of state transformations (i.e. matter, force/energy, information, and environmental conditions) are also used to describe the interactions between subfunctions.

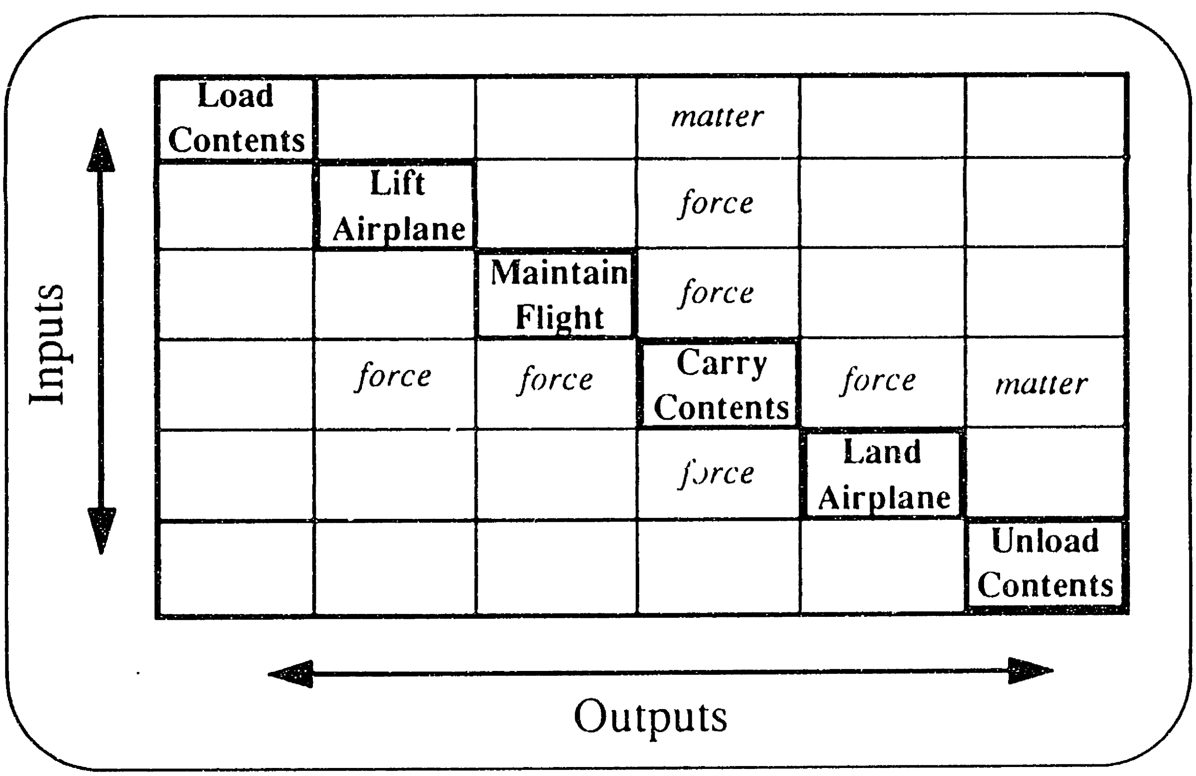

Figure 4.8: An N^2 Diagram for the Subfunctions of Transport Contents

The N^2 diagram provides two important advantages. First it lays out a rigorous process for checking the inter-relationships between each and every subfunction. Then it provides a visual display of both the big picture and details at once, allowing us to see the big picture while we concentrate on the details.

In developing the interactions between functions, it is helpful to group them into two categories: *fundamental interactions* and *incidental interactions* (Ulrich & Eppinger 144). Fundamental interactions are the desired and intended interactions between functions and the incidental interaction are the unintended interactions, which often are undesirable. In this example the transfer of force from Carry Contents to Lift Airplane is an example of an intended interaction -- the contents exert a downward force and the airplane is designed to overcome that force and lift the contents. In fact, overcoming the gravitational force exerted by the contents becomes the primary design objective of Lift Contents and is listed as such in the intrinsic characteristics for Lift Contents.

However, in lifting the contents, a reactionary force will be exerted on the contents of the airplane. Therefore, the contents will have to withstand the force required to lift them. Since this is an additional requirement placed on the contents, and not one of primary interest, it is considered an incidental interaction. As such it forms a design constraint for the function of carry contents and will need to be noted as such in the intrinsic characteristics of Carry Contents. As a constraint it will guide the design of the structure of the airplane to withstand the transmission of the lifting force for the lifetime of the airplane while also guiding the design of the restraint systems for holding the contents in place during take off.

In this way, the interactions of the functions, drawn out through the N^2 Diagram, form the invisible linkages between the branches of the function tree. The primary interactions will then be captured and clearly depicted in the functional schematic and the incidental interactions will show up as design constraints on the list of intrinsic characteristics.

Level Check

The fundamental principle of a functional hierarchy is that functions at the same level are *independent* of one another: While they do interact with one another in performing the higher level function, they do not contribute to the performance of one another. The difference between interacting and contributing is subtle, but significant. A function interacts with another function when the output of one function forms an input to the other. A function *contributes* to the performance of another function when it in fact performs a portion of that function.

For example consider the function of *brew coffee*, given the notion of a drip coffee maker. *Grind coffee* is a function that interacts with brew coffee in supplying an input, ground coffee, to the coffee brewing process. *Boil water*, on the other hand, is a function that contributes to the performance of brew coffee, since a drip coffee maker boils the water and passes it through the ground coffee in the process of brewing it; therefore, boil water is considered a subfunction of brew coffee.[10]

We ask the following questions for each pair of functions in the N^2 Diagram to test functional level:

> Does *Function A* contribute to the performance *Function B*?
>
> > If so, A is a subfunction of B.
> >
> > If not, does Function B contribute to the performance of Function A?
> >
> > > If so, B is a subfunction of A.
> > >
> > > If not, A & B are independent functions, and can reside at the same level.

---

[10] This example further emphasizes the importance of having a notion of the potential physical element when defining subfunctions. For example, if the physical concept was a French press rather than a drip coffee maker, both the ground coffee and the boiled water would be inputs to the process. Therefore in that case, boil water would not be a subfunction of brew coffee; however, depress plunger would.

After all the pairs have been tested, and neither A nor B are found to be subfunctions of any of the other functions, then A and B are confirmed to reside at the same level of the function tree. They are then considered to be independent subfunctions of the same higher-level function.

Group Functions

The *group functions* step is another check built into the process to help rationalize the list of subfunctions. In order to prevent the functional decomposition from spreading-out too wide, and becoming unwieldy, we try to develop between three and six subfunctions as a rule. The tendency is to generate more than six subfunctions, which often leads to groups of subfunctions at varying levels of detail. In keeping the number of subfunctions small, we reduce the risk of mixing levels and are also able to better maintain a system-wide focus.

First and foremost, the decision to group functions must be based on a rational conception of how the system will operate. In addition, another technique for grouping is based on an analysis of the N^2 Diagram. In analyzing the N^2 Diagram, one should check for functions that have both the same outputs going to a common function and the same inputs coming from a common function. If two or more functions have the same inputs and outputs and also appear to be strongly related, then they are good candidates for grouping..

In Figure 4.8, we analyze the N^2 Diagram shown in Figure 4.7 to check for potential groupings. In doing so, we discover that Lift Airplane, Maintain Flight, and Land Airplane all have the same output (a fundamental force interaction) going to Carry Contents. They all have the same input (an incidental force interaction) coming from Carry Contents. Since, those three functions do seem strongly related to one another and have the same functional interactions, we chose to group them in the function Provide Flight [F 1.3].

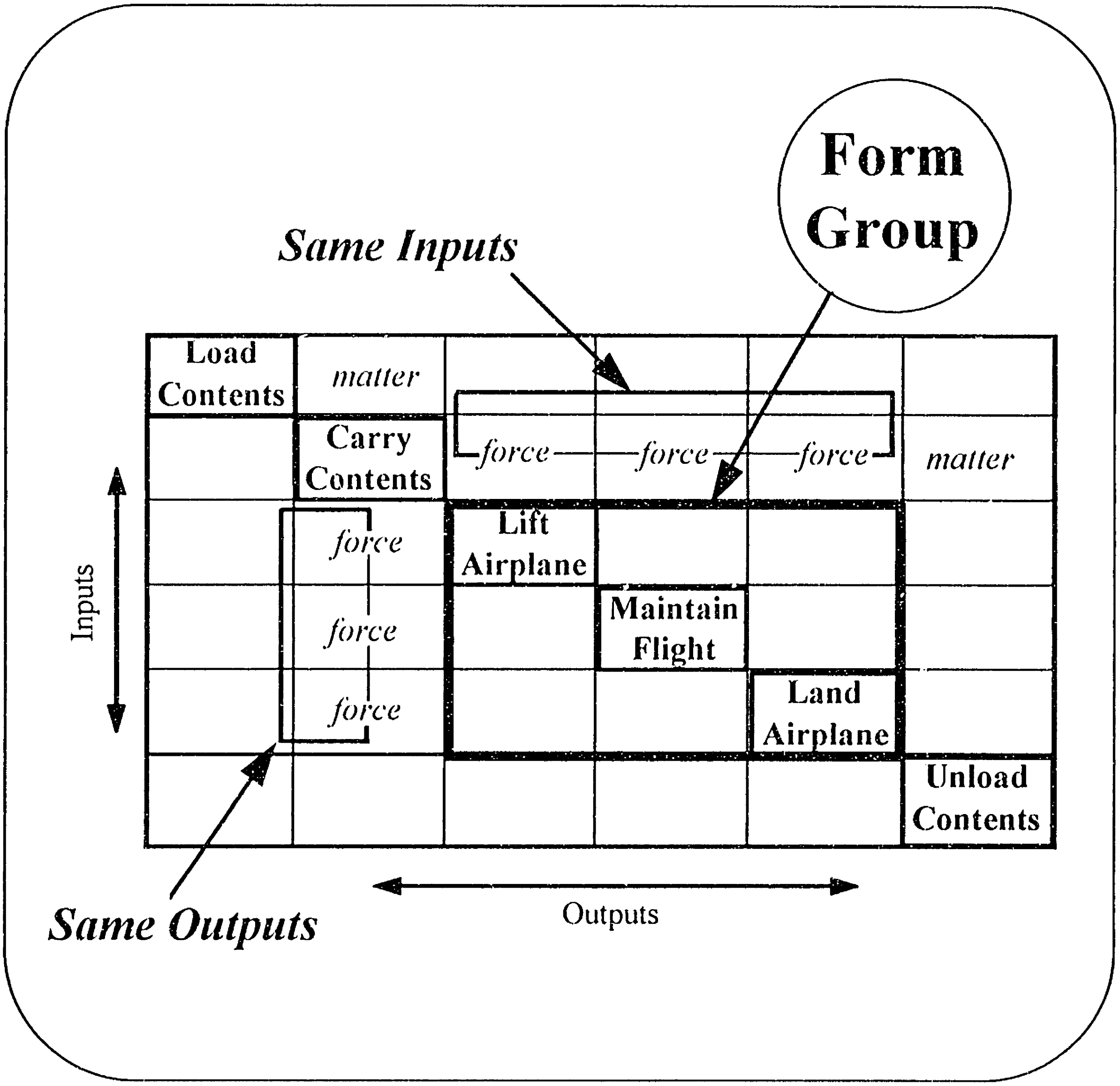

Figure 4.8: Use N^2 Diagram to Look for Possible Groupings of Functions

Functional Schematic

After completing the functional interactions, level check, and groupings; the N^2 Diagram contains so much information it is difficult to read at a glance. So once again we try to simplify the problem to prevent getting too bogged down in detail. In doing so we focus on the primary problem and create a separate diagram just depicting the fundamental interactions. The result is a functional schematic of sorts. A very basic one, for the subfunctions of Transport Contents, is depicted in Figure 4.9.

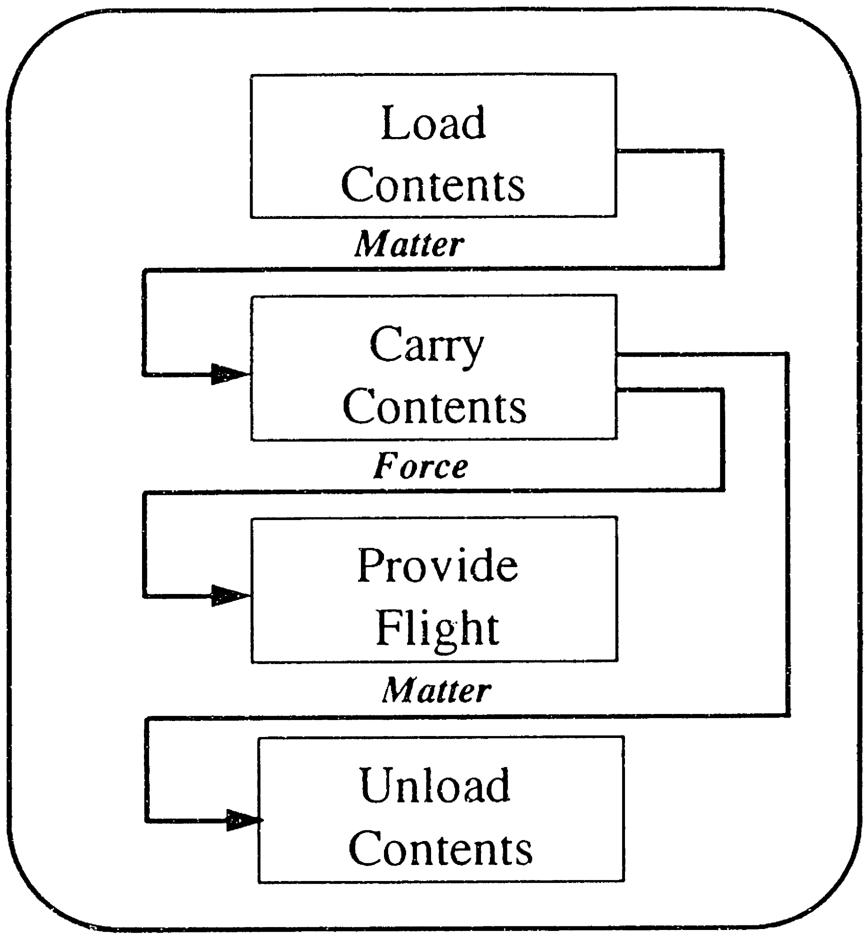

Figure 4.9: Functional Schematic

Fundamental Interactions of the Subfunctions of Transport Contents

IV-F) Identify Potential Physical Elements

Given a fairly in depth understanding of those subfunctions, we are then able to generate concepts for subsystems and components that can provide the required functionality. Our approach to this activity was to consider potential physical solutions for all the all the possible combinations of subfunctions. We found that in brainstorming physical elements to satisfy the subfunctions, we wouldn't ordinarily come up with elegant physical solutions that efficiently satisfy the functionality of several subfunctions at once. Therefore, we structured our brainstorming to briefly consider every possible combination of subfunction (e.g. singles, pairs, groups of three,...the entire set of subfunctions) and to generate ideas for their combined physical solution. After we have generated ideas for all the possible combinations, we then identify the working principles, basic physics, and technology associated with each potential concept.

A simple example of this can be seen for Transport Contents. We were able to conceive of a single Boarding/Loading Device system to accomplish both the Load Contents and the Unload Contents subfunctions. Since this physical solution was eventually selected in our simulation process, we chose to group the two functions together into the function Load & Unload Contents [F 1.2], as depicted on the function tree.

IV-G) Determine Variability, Product Architecture, and Physical Layout

In this step, the issues associated with transforming the functional design into the physical realm are analyzed. Up until this point, the functional approach advocated in this thesis has been pure, almost minimalist, in focus. While this minimalist approach is crucial for providing the design team with a strong focus on their primary mission, it does exclude the many issues surrounding the transformation of the functional design into physical components. Since these issues will impact the performance of the product in many subtle and substantial ways, they must be considered as well. Therefore, we address the question of how the physical reality will impact the functional design, before the physical concepts are finalized and selected.

Anticipate Future Variability

We differentiate between items that will never change during the product lifecycle and components that will vary over time or between customers. The objective here is to isolate the variability in order to provide maximum flexibility for the customers at minimal impact to the rest of the product. This can be achieved by designing the product with the knowledge of what is expected to change and what is expected to remain stable. The design must then be shaped around the anticipated variability of the product.

This is definitely a case of "a stitch in time saves nine." In taking into account issues of variability and architecture initially, the number of changes required later on will be greatly reduced. There are pitfalls associated with considering too much information up front; however this approach tries to structure the information in order to organize and not to overwhelm.

In differentiating between non-changing elements and variable elements, we first consider whether the function itself will be demanded by all customers throughout the life of the product. Then we look at it from the physical perspective: will all customers want this functionality satisfied the same way? Finally, we set up design guidelines to isolate the variability in order to provide the customer flexibility without incurring additional design engineering or manufacturing costs.

Figure 4.10 shows the variability assessment for the subfunctions of Provide Hand Cleaning Facility [F 1.1.1.5.5], which is located on the fifth level of the function tree. In the case of Provide Cleansing Agent [F 1.1.1.5.5.4], its functionality is permanent: as long as airplanes contain lavatories there will be a need to provide cleansing agents. However, the form in which the cleaning agent is provided will be a matter of the airline customer's preference. Since, airlines distinguish themselves based on the services they provide, they often concern themselves with these details; therefore, it is important to allow them the flexibility to specify the brand as well as the form (i.e. bar, liquid, or powder). As a result, a decision guiding the design was made to allocate a permanent zone for the Provide Cleansing Agent function while anticipating variability within that zone.

'Provide Hand Cleaning Facility'	Product Variability Considerations	Product Architecture Considerations
Provide Wetting Agent	Functionality: Store & Dispense are stable Transport may be variable (assuming a water system) to enable Lav. Flex. Zones	- Wetting agent itself is modular. - Support: leak & clog repair: modular; robustness: integral
Provide Cleansing Agent	- Functionality is stable - Airline Customer Choice of form, brand. - Allocate a zone for the function (within which there may be variability)	- Cleaning agent itself is modular. - Build: located on CRES counter-top - Support: refill & clean reqts: modular
Provide Hand-Washing Station	- Functionality is stable (may change in the long term?) - Appearance, size variations among customers	- Build: Integral with CRES counter. - Support: no modular reqts; robustness: integral

Figure 4.10: Product Variability & Product Architecture Considerations

Determine Product Architecture

In its most basic sense, product architecture can be thought of as the way a product is divided up into subunits, or chunks, and how those chunks relate to each other. If the interface between one chunk and all of the surrounding chunks is very discrete and the chunk can easily be replaced by a similar but not identical chunk, then that chunk is said to be ***modular*** with the rest of the product. However, if the interface is more pervasive and the chunk can not be replaced by other similar but not identical chunks without requiring significant changes to the other chunks, then that chunk is said to be *integral* with the rest of the product. While some products are purely integral (a monocoque body) and others are purely modular (a Lego's creation), most products will exhibit some combination of modular and integral architecture. For further discussion on product architecture, see Appendix B.

It is clear that the choice of architecture can have a significant impact on the product design. Furthermore, the choice of product architecture can have a significant impact on the performance of the product. Since most physical element concepts contain implicit product architectures, selecting the physical element concept is tantamount to selecting the product architecture. Therefore, the overall implications of product architecture are considered before choosing the physical element concepts, to ensure that the selected concept will not lock the product design into a suboptimal architecture.

Product architecture has implications for several aspects of the product beyond its functional design. Product architecture influences the design process; the build process: assembly and fabrication; the support process: maintenance and repairs; and the associated business processes: sales, purchasing, and material handling. Appendix B contains a table relating how these factors may influence the decision of where a chunk should reside along the integral-modular architectural spectrum. The distinction between variable and stable elements is really just a special case of considering architectural implications. If something will be variable, it should be modular with respect to the rest of the product. If something will be stable in design, it could be incorporated into a more efficient integral architecture.

Figure 4.10 depicts an analysis of product architecture considerations made for the Provide Hand Cleaning Facility function. In looking at the different architectural requirements associated with fulfilling the Provide Cleansing Agent function, we considered how the function will be performed as well as the associated design, build, support, and business processes. In terms of the functionality of a cleansing agent, it will be modular with respect to the rest of the product since it is removed in its use. Furthermore, we know that the cleansing agent is located on the counter near the sink based on the previous layout. The sink and counter will be made from a single sheet of stamped corrosion resistant steel (CRES), as decided during the previous physical concept selection. Therefore, the Provide Cleansing Agent function will be physically located near an integral component which is expensive to change. Finally, the support process for Provide Cleansing Agent requires refilling and cleaning between flights. Taking all these factors into consideration, a good architectural solution would be to design a system for providing the cleansing agent that is modular with respect to the steel counter. In doing so, the support process can be performed off the airplane in order to minimize airplane turnaround times, a strong customer demand.

Physical Layout

This step involves developing preliminary layout drawings for all the promising, feasible combinations of potential physical elements generated earlier. In creating the layouts, the team takes into account the anticipated variability and product architecture requirements developed in the previous steps. In so doing, this step serves to further develop the feasibility of the individual concepts before any one is chosen. Thus, it contributes to minimizing design rework by improving the information available for up front decisions.

IV-H) Concept Selection

All of these previous steps lead up to the selection of the best physical system to satisfy the functionality of the given function. The tool we apply to guide the selection process is often referred to as a Pugh Concept Selection Matrix, a simplified example of which can be seen in Figure 4.11. Since the Pugh concept selection process has been discussed in detail and proven in practice, it will just be described briefly in this section.[11]

[11] The Ulrich & Eppinger and the Clausing texts listed in the Reference section are good sources on the concept selection process.

Concepts / Criteria			*Reference* 	
Cost	-	+	0	+
Payload	-	0	0	-
Range	+	0	0	-
Speed	0	-	0	-
Safety	0	-	0	+
Total				

Figure 4.11: Pugh Concept Selection Matrix

The Pugh concept selection process operates on the principle of ranking concepts against one another based on how well they achieve certain criteria. In the process presented in this thesis, the selection criteria are taken from the intrinsic requirements list. The criteria should be chosen in order to help distinguish the concepts; therefore differentiating criteria of high priority to the customers and stakeholders should be used. The scores can be further differentiated by including weightings to reflect the relative priority of individual criteria.

To assist the team in scoring the various concepts, a reference concept is included. The reference concept should be a familiar, successful product that the team would like to improve upon. In addition to serving as calculation tool to select the best concept, the team can also try to combine all the positive features of the various alternatives into an even better concept.

Finally the best concept is chosen based on its ability to satisfactorily meet the criteria and to do so better overall than any of the alternatives. The selected system becomes the chosen physical concept corresponding to the given function and is then entered on the physical element tree. The subfunctions are reviewed given the selected physical element, and are then added below the given function on the function tree.

By incorporating Pugh concept selection at this stage of the concept development process, the team will be selecting the best overall system, not just the best individual components. This is further reinforced by comparing concepts in the form of system layout drawings, as opposed to the more common practice of comparing individual component concepts. Rather than selecting concepts based solely on their individual working principles, the process shifts the focus to how the concepts work together as a system and fit within the given product.

IV-I) Identify and Combine Common Functions

This hierarchical approach to concept development, as guided by the function and physical element trees, will yield a very complete solution. Sometimes, the solution will be even too complete. That is to say, there might be a more efficient design solution than the one that falls out of this hierarchical approach.

Since the function tree is targeted to developing a complete solution rather than an efficient solution, we added a process step to provide a focus on design efficiency. At the end of each concept development cycle, the team shifts their focus from the particular system concept they were developing and looks at the overall product. When they enter the function and physical element on the tree (and possibly pencil in the subfunctions and subsystems), it is an opportune time to consider the commonalties among functions.

Furthermore, the team tries to identify functions that can potentially be satisfied by a common physical element.

An example of how this might be done is shown in Figure 4.12. At the sixth level down, there are two separate functions to Provide Water [F 1.1.1.1.1.1 and F 1.1.1.5.5.5]. This should then raise a flag to the design team to consider the actual similarity of these two functions. Since, the functional nomenclature is fairly vague, it is best to decompose the functions further to check their similarity. In figure 4.12 we can see that both "Provide Water" functions have three subfunctions in common: Store Water on Airplane, Fill Reservoir, and Propel Water From Reservoir. Therefore, the team should propose some physical solutions that satisfy these simultaneously in addition to systems that satisfy the functions individually. If the common solution is determined to be better for the overall product, then it is selected as the chosen concept. Figure 4.13 shows and example of this for a common elements of a Water Supply System. In this example, both the Galley/Food Service System [PE 1.1.1.1] and the Lavatory System [PE 1.1.1.5] share the Three-Tank Reservoir System [PE 1.1.1.1.1.1.1], the Potable Water Fill/Drain Lines & Panel [PE 1.1.1.1.1.1.2], and the Pneumatic Reservoir Charging System [PE 1.1.1.1.1.1.3]. This is yet another example of how a system-wide focus leads to optimizing the total product.

The potential for a common physical solution was fairly easy to spot in this example. However, as the design increases in detail and the product team grows in size, these possibilities will become harder to detect. A coding process to analyze new functions as they are added to the tree could be developed to identify similar functions. This coding process could be based on the nature of transfer or transformation the function brings about. Therefore, the transfer/transformation categories of matter, force/energy, information, and environmental conditions could be applied for the first level sorting process. Such a system would need further development in order to define appropriate lower level sorting categories. Ultimately, the decision as to whether or not to select a common physical solution will depend on feasibility, timing, and the judgment of the design team.

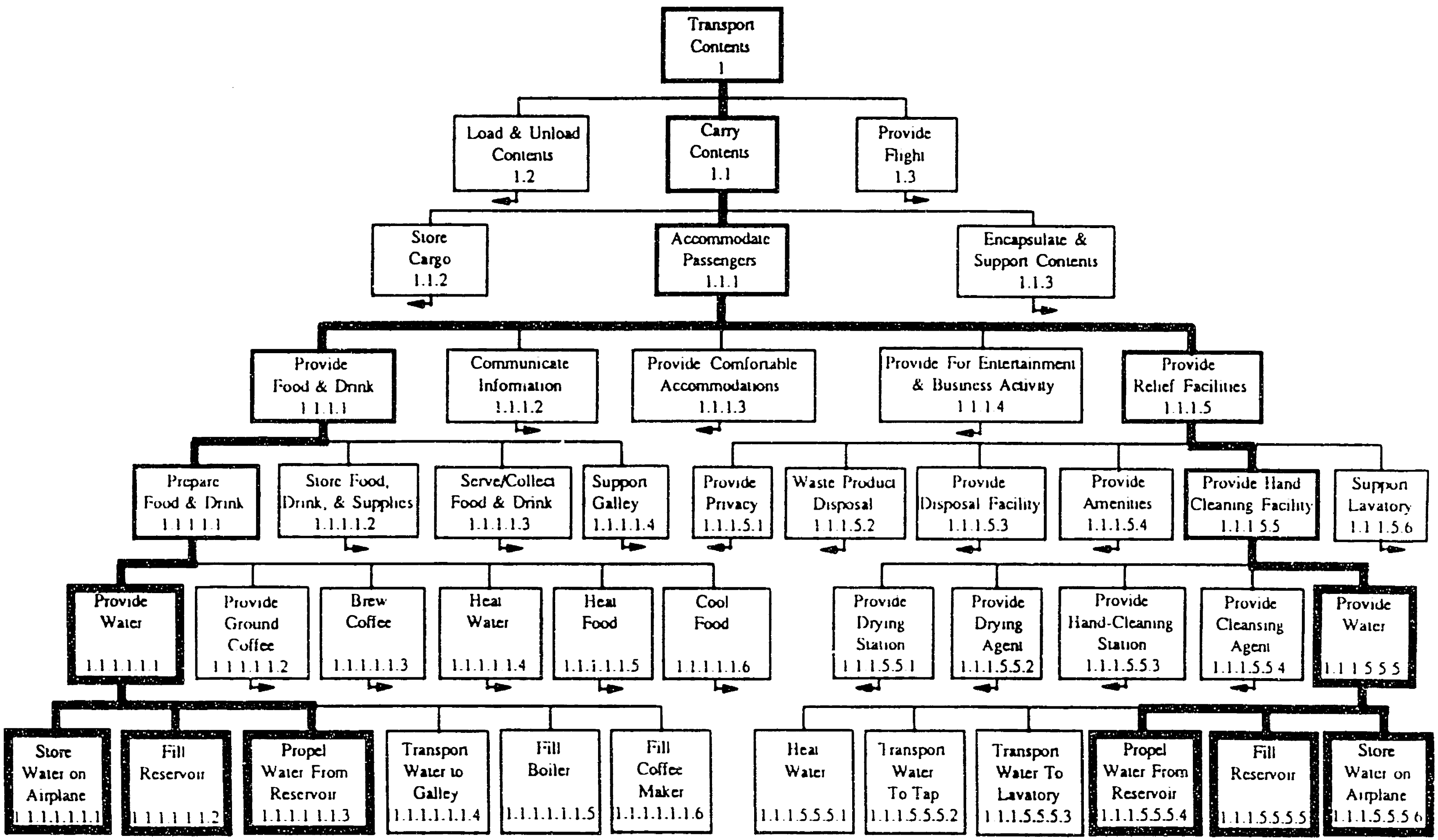

Figure 4.12: Function Tree as a Tool for Identifying & Combining Common Functions

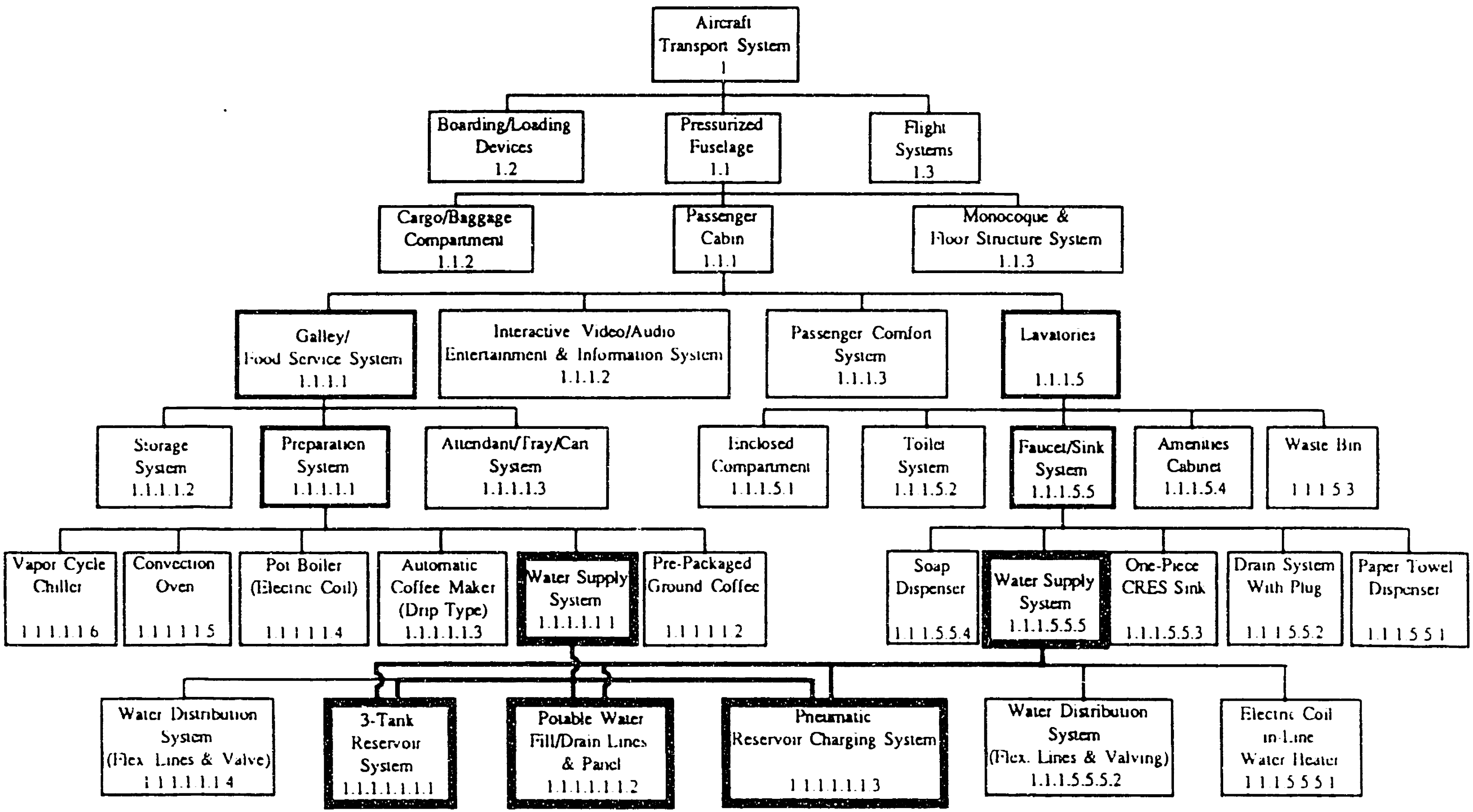

Figure 4.13: Efficiency of Design Through Common Reservoir System

V. Further Notes on the Process

V-A) Design Objectives & Guiding Principles

The approach to concept development described in this thesis brings many factors in to consideration early on in the process. While the contention is that this information is necessary to make better decisions, it does tread near the pitfalls of "information overload" and "analysis paralysis." The key to avoiding these traps is to form a very clear image of the product and service that you are striving for and to develop design objectives and guiding principles to keep you focused on that image

These types of meta-rules have often been employed to guide the design process. An example of one used at Boeing was conveyed by Robert Kelley-Wickemeyer of the Process Development Support Team. In that case, the design objective was to minimize the impact of safety hazards. To achieve that objective, the following guiding principles were employed: 1) Avoid the conditions that give rise to the hazard, 2) Alleviate the conditions that give rise to the hazard, and finally 3) Design to withstand the hazard. Another example developed during our process simulation is presented in Figure 5.1.

To be useful, the design team should develop a small number of objectives that both motivate the team and also guide decision making. Some examples of objectives that have been successful in the past include "Zero Defects" or "No Fasteners." These rules should go hand-in-hand with the image developed for the product and should reinforce that image. A few strong objectives, few enough for the team to keep them in the front of their mind during the entire process, will help focus the effort and enable the team to consider more information up front without getting bogged down during decision making.

Objective: Maximize Customer Flexibility

Guiding Principles:

- Isolate Variable Parts
- Maximize Determinate Manufacturing
- Maximize Parametric CAD Usage

Figure 5.1: Design Objectives & Guiding Principles

V-B) Handling of Support Elements

Support elements are those functions that support several other functions and also comprise a system of their own. In terms of their functional development, they arise as subfunctions in support of other functions and are then grouped together in an effort to promote design efficiency. The examples of support elements that we found during our simulation process include the physical structure of the airplane and components known as 'transport elements.' Transport elements consist of those items that transport matter, force, energy, and information across chunks and throughout the product. Examples of transport elements include electrical, hydraulic, pneumatic lines conveying force or energy; water lines and ventilation ducts carrying matter; and wiring carrying signals and conveying information.

These support functions serve both as subfunctions to various functions from the function tree as well as elements of the support system they comprise (e.g. the hydraulic system). Therefore they are handled differently from standard functions in that they are recorded twice on the tree. First they are recorded under the function they directly support (e.g. Support Galley [F 1.1.1.1.4] is located under the function Provide Food & Drink [F 1.1.1.1]). Then they are collected and organized under a separate branch of the function tree charting the functionality of the support system they are part of (e.g. Support Galley is also included as part of the structural support system contained in the function Encapsulate and Support Contents [F 1.1.3]).

The two listings of the function are then linked, because one can not be manipulated independently of the other: they are one and the same, just part of two different systems. Tracking and recording changes to these functions is an area in need of further development. A possible tool to facilitate these linkages might be in the form of an object oriented software system that enables the functions on the primary function tree to "point" to the underlying branches of the support systems.

V-C) Other Uses of the Function Tree

While the function tree is a very useful tool for developing the functional hierarchy of the product, it can also serve as a basis for organizing several other aspects of the design process. The function tree essentially drives the concept development process described in Chapter IV. Furthermore, the function tree can serve as a means for organizing the relevant information and for organizing the human resources, as will be described below.

The Function Tree as a Means of Organizing the Relevant Information

The function tree could serve as the organizing element for all the information generated during the concept development process, including the anticipated variability and the selected design, build, support, and business processes. Furthermore, the selected concepts recorded in the physical element tree could also be linked to their corresponding functions in the function tree. The mechanism for this linkage could be an objected oriented program where the function is the object and all the linked items are its attributes.

The Function Tree as a Means of Organizing Human Resources

Since the function tree spells out the types of functional disciplines required for the product program, it can serve as the basis for selecting the professional specialties to staff the program. Furthermore, the related functional schematic diagrams describe the interactions required between the functions and could thereby help lay out the channels of communication required within the product team. Thus, the function tree provides a means for organizing the human resources to carry the product development program.

VI. Conclusions & Recommendations

The concept development process described in this thesis offers many advantages over current design practice. While there are some obstacles to its implementation, they are far from insurmountable and are easily outweighed by the benefits of this approach. However, further research and development of this process is necessary to fully appreciate its full impact. The benefits, the obstacles, and the recommendations for further development will be described in the following sections.

VI-A) Benefits of the Process

Optimizes the 'Total System'

Applying a "system's thinking" approach to concept development is perhaps the most profound benefit of this process. Through the use of the function tree, the functional schematic, and other tools of systems engineering, the focus is strongly fixed on the total system. These tools and other aspects of the process enable more information to be considered in order to take into account the impact of these up front decisions on the total product and its associated design, build, support, and business processes. In this way, the overall product, instead of the individual elements, will be optimized.

More Efficient Process & Product

Rather than developing several different concepts well down in detail and then selecting the best one from among them, this approach generates several possible concepts at each level of detail. The one that best satisfies the requirements while complimenting the higher level concepts that have already been chosen is then selected. In doing so, this process provides the breadth of ideas but does so in a more efficiently by focusing further design effort solely on the best feasible solution.

In applying a system-wide approach, the likelihood of selecting incompatible solutions is greatly diminished. Therefore, the concept selection decisions will be of higher quality and costly design iterations will be minimized. As a result, design and production rework will be substantially reduced, leading to a more effective deployment of financial and human resources and again to a more efficient product design process.

Finally, the high level view of the functional design offered by the function tree provides the scope to seek out opportunities for satisfying several functions with a single physical element. In taking advantage of this opportunity, the approach described in this thesis favors the development of elegant design solutions and, thus, the development of more efficient products.

Better Handles Change

Many product change proposals today only consider some of the fundamental functional interactions and focus mostly on the physical interferences. As a result, changes are made that look good physically, but which introduce many phantom functional problems. Therefore, product changes often entail substantial test programs to try to catch and fix all these phantom functional problems. Furthermore, change proposals are accompanied by very inaccurate cost estimates, since the full impact of a change is rarely known at the outset.

In contrast, this approach can augment information on the more obvious physical impacts of the proposed change with the full extent of the functional implications. The up front work involved in developing the product functionality provides a source of reference information on all of the fundamental and incidental interactions between functions. This information will then help identify all of the functional implications attributable to a change. Therefore, in applying this approach, the design team will have a better idea of all the changes that will need to made in meeting the customer's new request.

In addition, this approach will give the design team a more accurate basis for estimating the cost and time required to make the desired change. It will therefore be easier to quantify whether the benefit the customer hopes to gain from this change justifies the cost associated with providing it.

This approach further improves on the change handling process by taking into consideration the anticipated variability of a product during concept generation. In so doing, concepts are developed that isolate the variable component so that they can be changed with minimal impact to the rest of the product. Thus, customers can be provided with the changes they desire at very little additional cost.

The Function Tree as an Organizing Mechanism

A final benefit of this process stems from developing the function tree. Since the function tree essentially drives the process, it provides an excellent structure for organizing all the information generated during the concept development process. Furthermore, the function tree also lays out the functional development of the chosen product concept. In so doing, it provides the ideal guide for organizing the product design team, based on the functions and their interactions. In designing the organization around the functional architecture of that specific product, the team will be structured to carry out the design process in the most efficient way.

VI-B) Obstacles

Time & Manpower to Develop

The major obstacle to applying this process is claim that excessive time and manpower will be required to consider all these downstream factors up front and to develop the product functionality. However, there are several counter arguments to this claim. The first is that these items must be considered at some point, therefore this process doesn't add work per se. In fact, it strives to reduce the total amount of work required by making decisions at the outset that simplify the downstream processes. Likewise, the functional development is something that many products require. In practice, they are usually "reverse engineered" during the testing phase, which can take as much time, if not more, than this approach for developing it initially. Furthermore, once the functionality has been developed, it can be reused in whole or in part for products with similar architecture.

Resistance to Formal Processes

A cultural barrier will also be faced in instituting any formal process for the creative activity of product design. However, as discussed earlier, this process is not a recipe to be followed by rote. It is a guideline for collecting the information to make better design decisions. It is a tool for channeling creativity along the most productive lines. This process requires individual thought to develop the ideas and tailor the process to the particular application. Finally, the process should not be imposed as a burden, but offered as an aid for carrying out the design of a complex manufactured products. When described in that light, it is difficult to argue that this process will constrict a designer's freedom.

VI-C) Recommendations for further development

Extending this Process Through to Detailed Design

During the development of this process, the question was raised regarding how this impacts the traditional notion of product design phases and if it can possibly be applied to detailed design as well. Product design is typically described as progressing from concept development to system design and finally to detailed definition. However, as people began to understand the power of this new approach, they began to question this traditional breakdown of product design into three steps. Essentially, this process applies the same approach to subsystem selection that it proposes for system definition, the only difference is the level of detail in the intrinsic characteristics that frame the problem.

The question of whether there is any difference in going from subsystem definition to detailed definition needs to be explored further. After all, isn't a component just a "subsystem," and a part then just a subsystem of a component? The only difference again is the level of detail. Therefore, shouldn't the same process apply?

Theoretically the same process could apply because a designer should still develop the functionality for a part even as detailed as a fastener. Furthermore, a designer should attempt creatively explore a variety of options for performing that joining function (i.e. a weld, a rivet, a screw, a snap-fit, etc.). Then, a designer would be wise to take into consideration anticipated variability and architectural implications (especially for assembly and service in this case). Finally, a designer should look at other functions on the tree to see if it would be possible for this one part to fulfill additional functions.

If detailed definition is simply a matter of targeting and refining specifications, would it not make sense to apply this approach to the entire design process? If refining specifications is a matter of finding the best form to achieve a function, subject to system constraints, then the answer is yes. Two different parts could be evaluated against each other based on how well they uphold those constraints (such as process finish, process capability, etc.).

The issues that need to be explored are how well the proposed process will perform at the very detailed level and what relative benefits it will provide there. This should be the subject of further process development research and process simulation.

Software Tools to Handle the Linkages

One of the benefits of this process involves utilizing the function tree as an organizing element for all the information generated during the concept development process. The most viable option for doing so would be to develop a software tool to handle the linkages. A mechanism suggested for carrying our these linkages could be an object oriented program where the function is the object and all the linked items are its attributes. The viability of such a software tool, and the implications of such a tool on the concept development process, require further exploration.

VI-D) Summary

The concept development process proposed in this thesis builds on existing research in the field of product development and systems engineering. It is in the bringing together of these two disciplines that this process ventures into new ground. In addition, several innovations were made during the process simulation activity. The end result is a process that is guided by the philosophies of function-driven, top-down, and system-wide focus to analyze more information during concept development. This approach leads to optimizing the total system, developing a more efficient process and product, and to handling change more competently.

After the simulation was completed, the participating engineers, who have contributed to many successful product design programs over the span of their careers, saw many advantages to this approach. Furthermore, those involved in estimating the cost of product development activities believed that this process will be significantly more efficient than today's practices. In order to fully appreciate the benefits of this process and to further understand its implications, additional research in applying this process to the detailed levels of concept development is necessary. Further research may demonstrate that this approach can be successfully extended beyond concept development into detailed design.

References

Alexander, Cristopher, *Notes of the Synthesis of Form*, Harvard University Press, Cambridge, MA 1964.

Anderson, Lindsay N., "Assembly Process Development for Commercial Aircraft Using Computer-Aided Tolerance Analysis Tools," MIT Leaders for Manufacturing Program, Masters Thesis, May, 1993.

Burchill, Gary, et al, *Concept Engineering: The Key to Operationally Defining Your Customer's Requirements*, Center for Quality Management, Cambridge, MA, Document No. 71, 1992.

Clark, Kim B., "The Interaction of Design Hierarchies and Market Concepts in Technological Evolution." *Research Policy* 14 (1985) 235-251.

Clark, Kim B., and Takahiro Fujimoto, *Product Development Performance*, Harvard Business School Press, Cambridge, MA, 1991.

Clausing, Don P., *Total Quality Development*, ASME Press, New York, 1994.

Corrigan, Robert E., and Roger A. Kaufman, *Why Systems Engineering*, Douglas Aircraft CO./Fearon Publishers, Inc., Palo Alto, CA, 1966.

Hall, Arthur D., III, *Metasystems Methodology: A New Synthesis and Unification*, Pergamon Press, Elmsford, NY, 1989.

Hauser, John, and Don Clausing, "The House of Quality," *Harvard Business Review* 3 (1988) 63-73.

Henderson, Rebecca M., and Kim B. Clark, "Architectural Innovation: The Reconfiguration of Existing Product Technologies & the Failures of Established Firms," *Administrative Science Quarterly* 35 (1990): 9-30.

Nobe, C.R., "On Defining Functions" The Boeing Company, Systems Engineering Technology (unpublished White Paper), Seattle, WA, 1994.

Phadke, Madhav S., Quality Engineering Using Robust Design, AT&T Bell Laboratories/P T R Prentice-Hill, Inc., 1989.

Proctor, Paul, "Boeing Refines Designs for 600-Seat NLA," *Aviation Week & Space Technology* 21 Nov. 1994: 48-53.

Ramaswamy, Rajan, and Karl Ulrich, "Augmenting the House of Quality with Engineering Models," *Research in Engineering Design* 5 (1994) 70-79.

The Seattle Times, "An Endeavor of Superlatives," 3 April, 1994.

Senge, Peter M., The Fifth Discipline: The Art & Practice of the Learning Organization, Doubleday, New York, 1990.

Shiba, Shoji;Alan Graham, and David Walden, A New American TQM, Center for Quality Management/Productivity Press, Cambridge, MA, 1993.

Sontow, Karsten, and Don P. Clausing, "Integration of Quality Function Deployment with Further Methods of Quality Planning," Laboratory for Manufacturing and Productivity, MIT (LMP-93-005: Working Paper), Cambridge, MA, 1993.

Suh, Nam P., *The Principles of Design*, Oxford University Press, New York, 1990.

Sutcliff, P. L. "Systems Engineering on the 777 and future Boeing Commercial Airplanes," The Boeing Company (unpublished internal report), Seattle, WA, 1992.

Ulrich, Karl T., "The Role of Product Architecture in the Manufacturing Firm," Research Policy 00 (1994), 1-22.

Ulrich, Karl T., and Steven D. Eppinger, *Product Design and Development*, McGraw-Hill, 1995.

Wiloughby, Willis J. Jr., Best Practices: The Transition from Development to Production, Department of the Navy, Washington, D.C., 1986, NAVSO P-6071.

Appendix - A: Examples of Intrinsic Characteristics

Transport Contents [F 1]

<u>Airline Customer</u>
Cost - $X Purchasing Cost
- $Y Operating Cost
Payload - # of seats
Range - *n* Nautical Miles
Speed - Mach *m*
Operates within air travel infrastructure:
- *Landing impact force*
- *Runway length*
- *Enplaning/Deplaning*

Range of Operating Conditions:
- *Altitude*
- *Weather*

Appearance:
- *Surface finish*
- *Subjective appearance*

Life Expectancy
- *Years*
- *Miles*
- *Take-offs/Landings*

<u>Corporate Strategy</u>
Time to Market - *# of months*
Parts Commonality
Supplier/Partner Roles

<u>MFG/Distribution/Servicing</u>
Dimensions
Packaging
Tolerences
<u>Safety/Ergonomics/Latent Needs</u>
Operational Safety
Environmental Safety
<u>Gov't Regulations</u>
FAA Regulations
EPA Regulations
Noise - *Stage 3 - x dB*

Carry Content [F 1.1]

Intrinsic Characteristics

• Costs:
- Ownership costs associated with Carry Contents
- Operating costs associated with Carry Contents

• Payload:
- Nominal value: *N* seats
- Range: *N* ± 75 seats
- Single, dual, tri-class configurations
- Total required passenger volume *X* $ft.^3$
- Revenue cargo & baggage capacity *X* Tons
- Revenue cargo & baggage volume *X* $ft.^3$

• Airport Infrastructure Requirements:
- Enplaning/Deplaning passengers
- Loading/Unloading cargo

• Range of Operating Conditions:
- Altitude
- Outside Temperature

• Appearance:
- Interior
- Exterior

• Life Expectancy:
- *N* years in service

• Corporate Strategy:
- Time to market - # of months
- Parts commonality
- Supplier/Partner roles

• Interior Flexibility:
- Pre-delivery: customized design
- Post-delivery: transformable product

• Design Functionality:
- Structural requirements for fuselages

Carry Content [F 1.1]

Intrinsic Characteristics
(continued)

- Processes:
- Design processes
- Build processes
- Support processes
- Business processes

- Duration:
- Flying time: 1 to 14 hours

- Turnaround Time:

- MFG./Distribution/Servicing
- Dimensions
- Packaging
- Tolerances

- Safety/Ergonomics/Latent Needs
- Operational Safety
- Environmental Safety
- Corporate Design & Performance Standards

- Gov't. Regulations
- FAA Regulations
- EPA Regulations

Provide Hand Cleaning Facility [F 1.1.1.5.5]

Intrinsic Characteristics

Cost: Ownership costs
Operating costs

Cleaning Performance Requirements: Per hand cleaning:

- water - 1 Cup
- soap - Cleansing-power-units
- towels - Absorption - X
  Surface Area - X

Capacity: (Usage Rates)

1st Class	-	3-4 washings/hr/lav
B Class	-	6-8 washings/hr/lav
C Class	-	12-16 washings/hr/lav

Class	Wash Rate	Water/ Cleaning	Flt Time	# of Lavs	Water Cap Req.
1st	3-4	1 Cup	14 Hours	2.5	9 Gal
B	6-8	1 Cup	14 Hours	4.5	32 Gal
C	12-16	1 Cup	14 Hours	12.5	175 Gal
					215 Gal Total

Range of Operating Conditions

Outside: Altitude: 41,000 ft
Delta Pressure: 9.1 psi

Inside Temp: 65°F - 175°F (Total Ambient Range)

Water Temp: 40°F - 120°F

Corrosion Requirements: Types of Corrosive Exposure

Variability Requirements: None

Provide Hand Cleaning Facility [F 1.1.1.5.5]

Intrinsic Characteristics
(continued)

Sink Height	Anthropometric Bases Reqt's. Min & Max.... Percentile coverage *X*% (could change by mkt.)
Sink Size	Volume - *X* Perimeter - *X*
Ergonomics:	IntuitiveControl.Conventions for: hot, cold, open, close... Meet req'ts. for persons with disabilities: one hand operation, etc.
Appearance:	Seamless
Life Expectancy	10 Years
Turnaround time:	Servicing within *X* minutes Cleanability - no sharp corners, no seams
Time to market:	Suppliers tooling capability
Structural Loads:	300 lb. "drunk load" Flight Payloads:
Static Loading:	No yielding under the following forces: Up *x* G Down *y* G Side *m* G Fwd *n* G
Dynamic Loading:	*N* G net forward (crash landing safety requirements)
De-lethalization:	No protrusion that could puncture vital organs

Health Requirements:

Water Purity:	"Potable"
Soap:	Hypoallergenic
Towels:	Sanitized

Appendix - B: Product Architecture

In the *Research Policy* Article "The Role Of Product Architecture In The Manufacturing Firm," Karl Ulrich defines product architecture in the following way.

> Product architecture is the scheme by which the function of a product is allocated to physical components....[or] more precisely as: (1) the arrangements of functional elements; (2) the mapping from functional elements to physical components; (3) the specification of interfaces among interacting physical components.

During the process simulation activities, and for the purposes of this thesis, we expanded the definition of architecture beyond the realm of "the mapping from functional elements to physical components." In doing so we adhere to the definition as stated above, but also consider the way the physical components are designed, how they come together in the assembly process, and how they are handled in the service and business processes (see Figure A2.1). Just as the overall product is broken up into physical units, or *chunks*, for the purposes of carrying out product functionality, so to is the product subdivided into chunks during the processes of design, assembly, and service. Additionally, the product is subdivided into chunks for other business related processes, such as option packages offered to customers, inventories of parts and subassemblies, and ordering and sourcing of parts. While many of these chunks will be defined according to the same boundaries, there will be cases when the definitions will be quite different. For example, a subsystem could be designed as unit and then be assembled from several chunks that are outsourced to different suppliers. In that case the product architecture guiding the design process will be different than the product architecture for the build process.

In article cited above, Karl Ulrich describes two types of product architecture: *modular* and *integral*. Those terms differentiate how different chunks relate to rest of the product. If the interface between one chunk and all of the surrounding chunks is very discrete and the chunk can easily be replaced by a similar but not identical chunk, then that chunk is

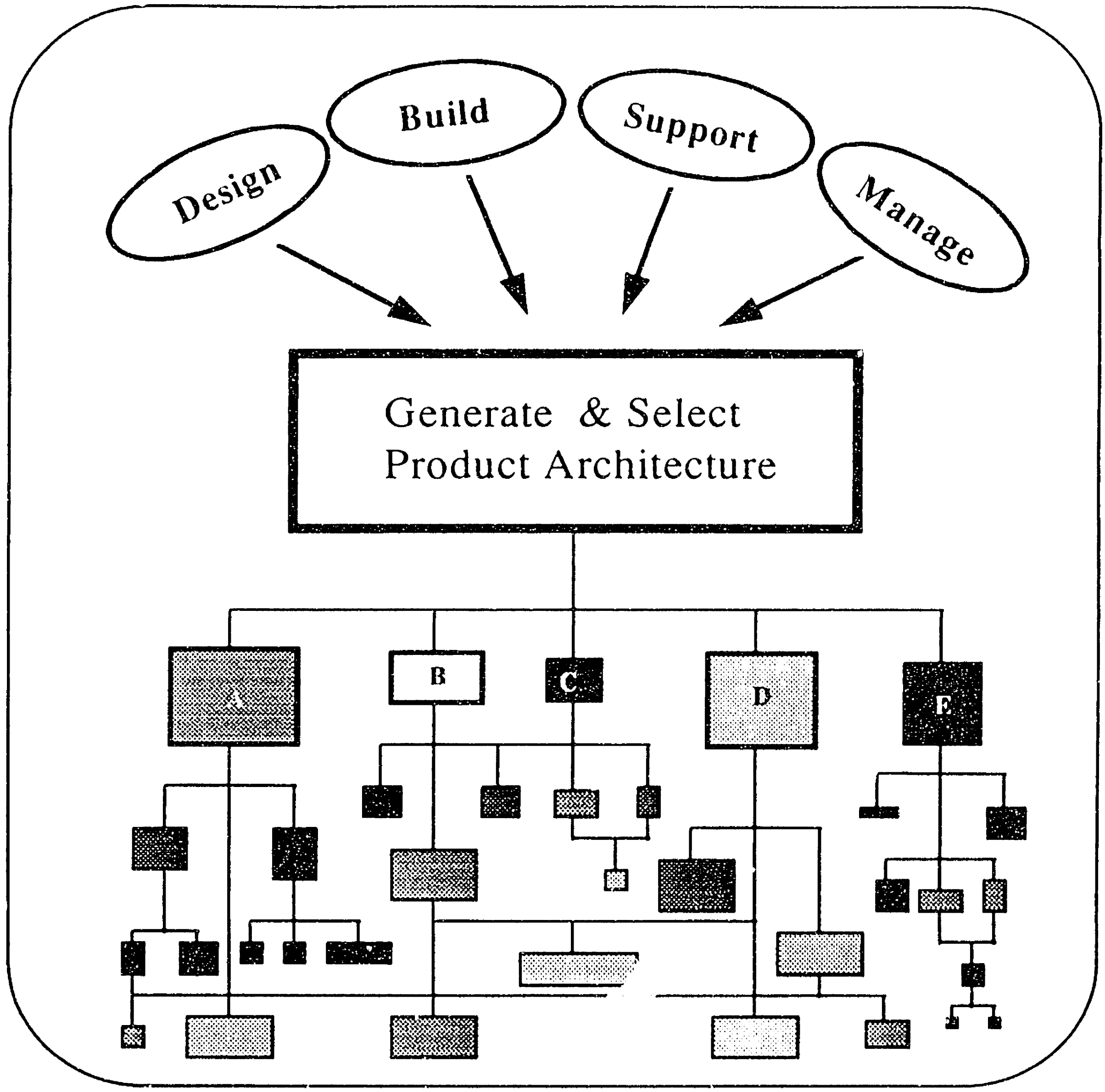

Figure B.1: Product Architecture - Contributing Influences

said to be modular with the rest of the product. However, if the interface is more pervasive and the chunk can not be replaced by another similar but not identical chunk without requiring changes to the other chunks, then that chunk is said to be *integral* with the rest of the product. Most products will exhibit some combination of modular and integral architecture.

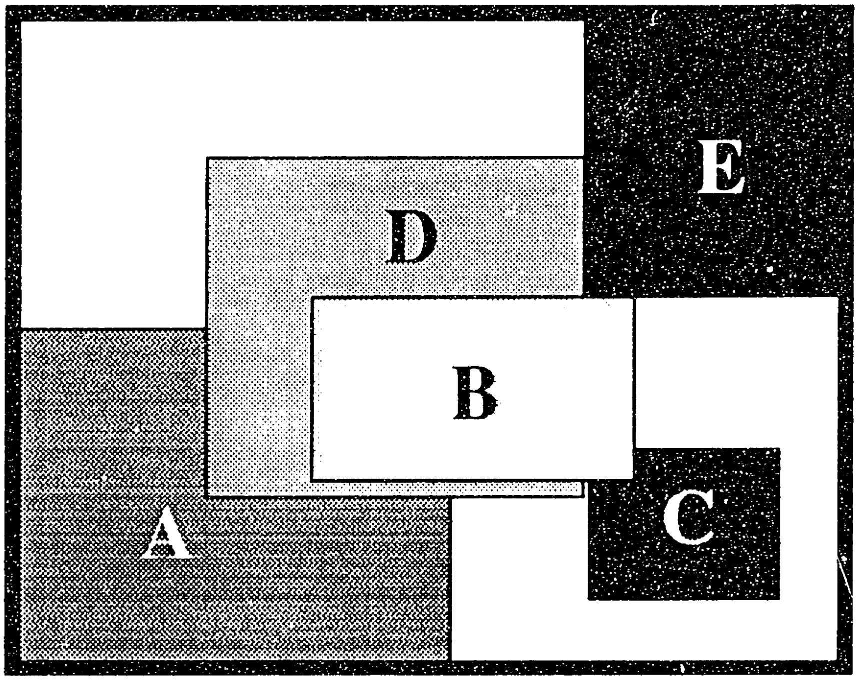

Figure B.2: Modular & Integral Chunks

In Figure B.2 the shaded blocks represent the product chunks for a hypothetical product architecture. When selecting a product architecture, one determines the boundaries and the degree of modularity of each chunk with respect to the rest of the product. In the above case, for example, five chunks of varying modularity were chosen. The degree of modularity can be described as the extent the design solution of one chunk constrains the possible solutions of the other chunks. In the figure above, this is demonstrated by the amount of overlap among the blocks. The less overlap, the more modular the chunk. The more overlap, the more integral the chunk. (The darker the block, in the above example, the greater its modularity.) Block E is almost completely modular: it only impacts the design of the rest of the product through a narrowly defined interface. Block B, on the other hand, is highly integrated with blocks A and D, and partially integrated with C. The figure above demonstrates the possible variety in the level of modularity that can exist within a product architecture. Furthermore, while block E is modular with respect to the rest of the chunks, in its own 'chunk architecture' it may, in fact, be highly integrated.

As mentioned earlier, different factors will exert different influences on the selection of product architecture. Table B.1 lists several factors along with a description of the form or forms of architecture (modular or integral) which best suits each factor.

Table B.1: Factors Influencing Architecture Selection

INFLUENCING FACTOR	FAVORED ARCHITECTURE
Function	
● Enhance product functionality	**Integral** in theory will optimize performance (since there are fewer external constraints), but **modular** --by breaking down the problem to more easily satisfied sub-problems--may result in better performance in practice.
◆ System optimization	**Integral**
◆ Flexibility in use	**Modular** (i.e. a transformable product)
Design	
● Increase engineering design efficiency	**Modular** in theory will optimize efficiency (since the big problem is broken up into easier to solve sub-problems). However, an **integral** architecture may enable a less complex design process as the number and extent of design interfaces grows large or the interfaces become less standardized.
◆ Product Change: - Isolate variable elements from non-variable elements.	**Modular**, for those elements that are expected to undergo design changes during the product's life-cycle.
◆ Design Commonality: - Part commonality across product families - Component standardization.	**Modular**, will enable both aspects if design commonality (whereas an integral solution provides a customized solution). However, design incentives/initiatives to share parts must be instituted, or the end result will merely be a proliferation of parts.
Build	
● Promote ease of production	**Integral**, will result in fewer parts and therefore fewer pieces to assemble. **Modular**, will result in more standard interfaces and therefore a less complex assembly process.
◆ Ease of manufacturability	**Modular,** will tend towards smaller, less complex parts which are generally easier to manufacture.
◆ Ease of assembly	It is a function of: • # of parts • # of operations • difficulty of operations 'If' reducing the number of parts will result in a very complex assembly procedures; 'then' use a **modular** approach; 'else' use an **integral** approach.

Table B.1(continued): Factors Influencing Architecture Selection

INFLUENCING FACTOR	FAVORED ARCHITECTURE
Support	
● Enable ease of maintenance	**Modular**, generally enables easier access and replacement of components. **Integral**, results in fewer parts and thus fewer part failures as a rule.
◆ Replacement of worn components	**Modular**, if parts are likely to wear. **Integral**, if wear can be avoided by fewer parts.
◆ Repair of failed components	**Modular**, if parts are likely to fail. **Integral**, if failure can be avoided by fewer parts
Business	
● Streamline business processes	**Modular** usually simplifies through increased standardization.
◆ Customer ordering	**Modular**, provides a standard list of options to choose from, plus quicker turn-around as no new design work is required. **Integral**, provides ultimate flexibility in tailoring the product to the individual customer's requirements.
◆ Supplier/partner interfaces	**Modular**, enables outsourced components to be "black-box" designs, only requiring coordination at the physical interfaces. **Integral**, may be a better solution where the number or complexity of interfaces makes the coordination task too burdensome. Then it may be better to bring the supplier/partner in-house to jointly contribute in an integral design effort.
◆ Inventory handling	**Modular**, may result in fewer part numbers to stock when combined with a parts commonality approach (but probably more physical pieces per finished product). **Integral**, generally leads to fewer separate components and therefore less parts to stock (but probably more part numbers).