AN ABSTRACT OF THE DISSERTATION OF

Robert Lewis Nagel for the degree of Doctor of Philosophy in Mechanical Engineering presented on June 10, 2010.

Title: A Design Framework for Identifying Automation Opportunities

Abstract approved: ____

Robert B. Stone

Since the Industrial Revolution, automated systems have increasingly become a part of everyday life. Automated systems range in scale and scope from simple assists to complete robotic replacements. Inexpensive integrated circuits, sensors and microprocessors have allowed automated systems to become smaller, cheaper and more prevalent. Identifying appropriate opportunities for automation, however, is still a complex task. This research aims to discover automation opportunities for the design of products that replace error prone human-centric tasks. Process models are used during conceptual design to capture how customers will use products. These process models combined with functional modeling provide a starting point for the systematic exploration of automation opportunities. Failure analysis based on process models is used to identify customer-product interaction points offering opportunities for automation. Impact factors qualitatively and quantitatively predict the impact of an automated solution as it relates to an identified failure mode.

The integration of functional and process models creates a framework to support modeling of manual processes. This framework is used for the discovery and conceptualization of products to fulfill identified automation opportunities and consists of the following four stages. (1) Identify and understand the needs of the customer. (2) Translate customer needs into engineering specifications. (3) Identify automation opportunities from the manual actions of the customer. (4) Synthesize solutions to the identified automation opportunities. This framework defines the underlying structure for a family of methodologies to automate existing manual products. One methodology, supported by the framework, is presented in this dissertation for identifying automation opportunities based on error prone human-centric tasks.

Designing an automated solution for identified opportunities has the potential to improve task completion efficiency, provide safer product operation, provide improved convenience and improve quality control with the final outcome of a task. Applications of this research include automated systems for military or defense applications, assistive technologies that improve quality of life, therapeutic devices for clinical applications, design of sustainable products, design of consumer-focused products and smart home technologies.

A Design Framework for Identifying Automation Opportunities

by
Robert Lewis Nagel

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented June 10, 2010
Commencement June 2011

Doctor of Philosophy dissertation of Robert Lewis Nagel presented on June 10, 2010.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial and Manufacturing Engineering

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Robert Lewis Nagel, Author

ACKNOWLEDGEMENTS

I acknowledge and thank my advisor, Dr. Robert B. Stone. Your support, encouragement and guidance throughout my degree have made my graduation from Oregon State University possible.

I acknowledge and thank my committee members, Dr. Irem Tumer, Dr. Robert Paasch, Dr. Karl R. Haapala and Dr. Milo Koretsky as well as research colleagues Mr. Kerry Poppa with whom I collaborated on Chapter 2 Background on Design Approaches, Mr. Joseph Donndelinger and Dr. Ryan Hutcheson with whom I collaborated on Chapter 4 Integrating Functional and Process Modeling, Dr. James L. Greer with whom I collaborated on Chapter 6 Process-based Failure Propagation and Impact Factors, and Dr. Daniel A. McAdams my co-advisor while a masters student at Missouri University of Science and Technology. Your input has helped to formulate and direct my path through this research.

I acknowledge and thank Mr. Kenneth Perry, the undergraduate researcher with whom I worked very closely on a number of projects at Missouri University of Science and Technology. Your coding skills proved to be an invaluable part of realizing the physical form of Chapter 4, FunctionCAD, as well as bringing MAX, our IGVC robot, to life.

I acknowledge my friends and family for your support and encouragement. Lewis and Linda Nagel, my parents, I appreciate everything you have done for me; your encouragement has driven me. Jacquelyn K. Nagel, my wife, you are my best friend, my confidant—I love you. My friends in the Design Engineering Lab, you keep every day interesting. Thank you.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure | Page

LIST OF TABLES

LIST OF APPENDIX FIGURES

LIST OF APPENDIX FIGURES (Continued)

LIST OF APPENDIX FIGURES (Continued)

LIST OF NOMENCLATURE

Automation: A system, methodology or technique aimed at replacing manual operations or control to reduce human interaction during a manual process ("Automation" 2001).

Automation Opportunity: A set of circumstances within a manual process appropriate for replacement by a system, methodology or technique aimed at providing full or partial operation and control of the manual process.

Black Box (Functional) Model: The high-level functional model defined by a single overall operation of the product being designed.

Black Box (Process) Model: The high-level process model defined by a single overall event representing the task to be accomplished.

Child Failure: A failure mode that results in a flow being faulted through direct interaction with the failed configuration chain.

Conceptual Design: The phase in the engineering design process when concepts are generated, evaluated and refined.

Configuration: A specific discrete instance of the overall function of the product which may relate to the environment where the product is used or specific applications of the product. The configuration of a product is modeled functionally.

Configuration Model: A detailed model of the individual actions and changes occurring to the product as a whole and involved in completing a particular event.

Configuration Failure: A failure mode that occurs when a single configuration change within an event can no longer occur. Configuration failures do not cause the entire event to fail, yet may degrade the performance of the event.

Core Functionality: The function-flow pairs that are essential to the most basic operation of the product. Often these are the functions supporting the operational event of the product. Function-flow pairs not considered core functionality are those related to events such as maintenance or storage as well as supporting functions.

Detail Design: The phase in the engineering design process when engineering and economic calculations are performed, final form, layout and dimensions are established, and documentation is completed.

Embodiment (Embody) Design: The phase in the engineering design process when physical form is applied to the chosen solution based on back-of-the-envelope style engineering and economic calculations.

Energy: The physical property that causes a change in work capacity of a technical system (Pahl et al. 2007). Mechanical and electrical are examples.

Environment: Aggregation of all conditions, things and influences that surround a system ("Environment" 2001). All product operations and interactions occur in an environment, and all other flows not included with the product must be derived from the environment.

Event: A set of configurations of a product, which pertain to changes to the operability of a product or sequencing of operations during the usage of a product.

Event Model: A more detailed process model consisting of multiple events that collectively define the customer's operations with the product.

Event Failure: A failure mode that occurs when an event within a process does not function properly, but the overall process remains operational.

Flow: A material, energy or signal, which interacts with the product; flows are expressed as nouns (Stone et al. 2000).

Framework: The structure necessary to methodically study an opportunity ("Framework" 2001). A framework provides the basic structure underlying a methodology.

Function: A description of an operation, expressed as the active verb, performed by an artifact, as a part of a larger product, to transform an input flow to a desired output flow (Stone et al. 2000). Functions are tied together via material, energy and signal flows. Also referred to as a Sub-function.

Functional Model: A structured arrangement of functional elements tied together via material, energy and signal flows describing an artifact or collection of artifacts that collectively comprise the product (Stone et al. 2000). Also referred to as a Function Structure.

Functional Modeling: The overall approach to modeling what a product must do in terms of elementary operations such that the product may achieve an overall objective (Stone et al. 2000).

Function Structure: See Functional Model.

Human-centric: A set of actions that focus on the human element (i.e., the operations of a system taken from the perspective of the human operator) as a part of a complex system.

Material: Any matter interacting with or traveling through a technical system (Pahl et al. 2007). Materials may converge, diverge or be changed through functional systems. Solids, liquids and gasses are examples.

Method: A procedure or a technique that provides a means to do something in accordance to a defined plan ("Method" 2001).

Methodology: A system of methods ("Methodology" 2001).

No Flow Failure: A failure mode that results in a function or a configuration that is no longer operational (Krus and Grantham Lough 2007). The flow associated with the function or configuration that failed also fails. The remainder of the flow chain is no longer operational.

No Transformation Failure: A failure mode that results in a function or configuration that is no longer operational. The flows through the function or configuration is no longer transformed, but otherwise the flows remain unaffected. The remainder of the flow chain continues to be operational.

Preliminary Design: The phase in the engineering design process when customers are identified, their needs are collected and the design problem is being understood.

Process: The set of defined events that occur with respect to the product as a whole and aim to meet a particular goal. Processes are tied together via the product, material, energy and signal flows.

Process Failure Levels (PFL): Approach to qualitatively assess the potential impact of propagated failures. Process Failure Levels are based on the hierarchy of process models.

Process Modeling: The overall approach to modeling a series of customer-driven, product-based operations related through input and output flows, the product being designed, and time.

Process Terminal Failure: A failure mode that ends a process completely. The process cannot be restarted. Process terminal is the most severe process-based failure mode.

Process Transient Failure: A failure mode that ends a current instantiation of a process. The process can be restarted at a future time.

Propagated Failure Analysis (PFA): Approach to identify and propagate failures from a human-centric perspective. Propagated Failure Analysis is based on hierarchical process modeling; failures are propagated through the configuration models abstracting a process.

Signal: The information, which is used to affect an action or to report on an action occurring as a part of a technical system (Pahl et al. 2007). Control and status are examples.

Sub-function: See function.

LIST OF NOMENCLATURE (Continued)

Supporting Functions: Those functions that describe the manufacturing, assembly and support features that are present in the embodied form of a product (Bohm and Stone 2004).

System: A structured arrangement of functional elements tied together via material, energy and signal flows describing an artifact or collection of artifacts. Functional models are often used for system representation.

System to Process Sensitivity (SPS): Approach to quantify the impact of propagated failures based on the loss of flow paths through a configuration model.

Time Line: A representation of the temporal relationship between events and configurations in a process model.

To my grandfather, Robert Charles Nagel.

A Design Framework for Identifying Automation Opportunities

CHAPTER 1 Introduction

Since the Industrial Revolution, automated systems have increasingly become a part of people's everyday lives. These automated systems range in scale and scope from simple assists to complete robotic devices. Some of the earliest forms of automation harnessed the power of steam to power factories and enable economical shipping—both on land and at sea. Initial control of these automated systems was manual, but quickly, automated controls such as flywheel governors and floats were invented (Sheridan 2002). Today, automated systems range in size and scope. They have entered many facets of our daily lives; they explore hidden corners of our planet and the edges of our solar system. And, with technology driving down the cost and size of circuitry, microprocessors and sensors, automated systems continue to become smaller, cheaper and more prevalent.

This growth, rapid change and acceptance of automation can be seen every day. For instance, consider a simple, mundane chore such as mowing grass. The first patent for a manual-powered, reel-style, lawn mowing mechanism was issued in 1830 to Edward Budding (Schroeder 1993). Following his patent came many other lawn mowing mechanisms, acceptance grew, and automation efforts continued to lessen operator effort. On November 13th, 1897, *Scientific American* published an article on the first gas powered, riding lawn mower, and in 1903, Thomas Caldwell developed a similar steam powered, riding lawn mower (Schroeder 1993). By the 1950s, technology allowed remote control (Schultz 1999), and now fully autonomous lawn mowers can be purchased from companies such as Husqvarna (Husqvarna 2009) and Kyodo (Kyodo America 2010).

Trends such as the one demonstrated by the lawn mower are evident in other areas. For example, in manufacturing, mechanized systems first replaced hand tools. Then, these mechanized systems were energized speeding the manufacturing process. Robotic systems now often replace or assist humans on the factory floor improving many factors such as safety, repeatability, reliability, and efficiency.

Entire fields and industries have grown around the idea of automation. For example, Mechatronic Engineering blends computer, mechanical and electrical engineering into a synergy where the sum of the parts are designed together from conceptual design forward (Bradley et al. 1991; Popovic and Vlacic 1999; De Silva 2005). For this De Silva recommends that a strong understanding of functionality is important for a successful design. The resulting designs are to be more efficient and give a performance boost over products where independently designed mechanical, electrical and computer constituents are connected during the latter phases of design (De Silva 2005). How to achieve this synergy is, however, far from settled. Recognizing this need for synergy, Middleton describes principles based on understanding processes to be automated as a starting point to develop an automated solution (Middleton 2000). Other methods advocate following an engineering design approach beginning with the initial customer needs (Bradley et al. 1991), and others yet focus on the energy control aspect of the design considering the mechanical and electrical energy flow through the system (De Silva 2005).

To further formalize the design of mechatronics, researchers, such as Chen et al. and Gausemeier et al., approach mechatronic systems from an engineering design perspective. Chen et al. have established a conceptual design approach that is based on traditional functional modeling methodologies. Their approach represents mechatronic systems functionally during the conceptual design stage (Chen et al. 2002; Jayaram et al. 2003). Further formalizing functional modeling representations of mechatronic systems, Gausemeier et al. develops a functional modeling language specifically applicable to mechatronic systems (Gausemeier et al. 2001). Functionally understanding a mechatronic system during conceptual design can help insure a truly integrated system since solution strategies may be mapped to functionality without considering the specific divisions of traditional engineering domains. These approaches, while being necessary steps to formalize the design of automated systems, do not address the question: Where should automation be implemented?

This evolution toward automated systems can come at a price. Negative impacts affiliated with automation can include issues related to trust, social contact and trade skills (Sheridan 2002). Sheridan, in his text *Humans and Automation*, discusses the negative and positive aspects of introducing automation. People may begin to over or under-trust automation systems. Physiologically, people may feel abandonment, boredom, stress or anxiety. Social contact is often lessened as people are replaced with robotic systems; skills tend to be lost; and accountability for problems may be abandoned (Sheridan 2002). These issues, first identified in manufacturing settings will spread as automated systems increasingly become a part of peoples lives.

During the design process, decisions must be made as to whether or not the negative impacts outweigh the positive aspects. Automation can provide safer work environments and safer product operation. Efficiency can be improved with less down time, improved repeatability and more reliable results. Faster response times can result from anticipated stimuli. Effort can be reduced; convenience can be improved; and work time can be reduced (Sheridan 2002; Pethokoukis 2004). Addressing these positive and negative aspects of automation should be performed during conceptual design to ensure a final product that meets the customers' needs.

1.1 Hypothesis

The underlying hypothesis of this research is that indicators may be identified during conceptual design to identify human-centric actions that are ripe for automation; these actions are termed *automation opportunities.* Following a systematic design approach, it is possible to understand the actions through which potential customers use products to complete a task. Process-based failure analysis allows the exploration of failures related to unexpected customer actions. These failures, used as opportunities for automation, seed the generation of concepts aimed at replacing human-centric actions with automated solutions. The result is a methodology through which automation opportunities may be discovered during the design of products for the replacement of error prone, human-centric tasks.

1.2 Motivating Applications

Identifying automation opportunities holds potential for applications in assistive technologies, military and defense automation, manufacturing and workplace, home automation and sustainability. Each of these motivating applications are discussed individually in the following subsections.

1.2.1 Assistive Technologies

Assistive devices are tools and equipment that may be used by persons with disabilities to improve overall functionality (U.S. National Library of Medicine and National Institutes of Health 2010). Automation may be developed based on the human-centric actions of non-disabled persons performing basic, daily tasks (e.g. employing mimicry) or based on the desired actions of persons with disabilities from a set of needs derived from interviews and ethnographic studies. Integration of automation into the lives of persons with disabilities may enable basic actions and operations mimicking normal human actions, increasing freedom and improving quality of life.

1.2.2 Military and Defense Automation

Automation, in the form of remotely controlled and autonomous robotics, will change the face of warfare. Currently, the United States has 5,300 aerial robotic systems and 12,000 ground robotic systems deployed and completing tasks such as ordinance disposal, surveillance, and tactical military strikes (Brown 2010). As acceptance increases, deployed robots will increase, the technology behind these systems will advance, and battlefield risks will decrease. Robots are capable of penetrating enemy borders to perform clandestine strikes and surveillance. They can replace human hands for explosive ordinance disposal and remote decontamination. Robotic systems will perform the tasks and operations, originally, rigidly defined by MIL specifications, and the lives of soldiers will be saved.

1.2.3 Manufacturing and Workplace

Manufacturing environments have perhaps changed the most dramatically with the development of robotics. It is in manufacturing where robotics have been most widely explored and adopted as a viable replacement for human counterparts. Much of the research on the negative and positive aspects of robot deployment relates to manufacturing applications (Sheridan 2002). And, it should be no surprise that much of the development of robotic systems has been driven by the manufacturing workspace. Replacing humans in a manufacturing facility is not only cost effective, but also a way to raise efficiency, increase safety, improve reliability, boost quality control and ensure repeatability (Pethokoukis 2004).

Beyond the manufacturing environment, robotic systems will begin to enter the office workspace. For example, researchers at Carnegie Mellon have developed a robot that replaces receptionists (Pethokoukis 2004); an idea that brings up many social issues, but none-the-less demonstrates how human-actions might be translated into automated systems. We do not have to look far for other examples. The United States Postal System's current plight (cancelation of Saturday mail delivery (Pearlstein 2010)) demonstrates how quickly paper mail is being replaced through e-mail systems, and in an office environment deliveries are often made through robotic mail carriers that follow a predetermined path through a workplace. Also, transactions and reservations may now be placed through computer terminals replacing what were once secretarial jobs, and as companies strive for the benefits associated with automation, these types of automation will continue to be commonplace.

1.2.4 Home Automation and Sustainability

Automation, such as demonstrated with the lawn mowing example, will continue to enter many facets of our daily lives. In the home, robotic vacuum cleaners and floor mops such as those by iRobot (iRobot 2010) will help with cleaning. Dishwashers and laundry machines clean our dishes and clothes. New devices will continue to be designed such as refrigerators that order food as supplies dwindle, microwaves that automatically prepare food for meal time, showers—or even whole bathrooms—that self-clean and disinfect. As the underlying technology improves, these devices will become smarter, more efficient and more reliable. Their continued development will focus on the jobs and tasks that arise in our daily lives. They will facilitate the completion of these tasks either through assists or complete replacement of their human counterpart, and in time, they will change how people interact with their homes.

The continued development of home automation will also advance the overarching goal of sustainability. Automated products would sense a person's entrance and adjust the lighting, heating, clocks, appliances to predefined settings. Instead of individually controlled devices, each electrical product in the home could be interconnected within a mesh network to allow remote activation/deactivation; when the home owner is away from home, the house would *sleep*. Other products may, however, be less automated providing nothing more than an indicator to cue the customer to follow more sustainable usage practices. And, as these products are developed, they will expand to schools, workspaces, and community buildings. They will mesh within peoples lives, replacing manual actions, automating home systems, and helping people to reach a more sustainable existence.

1.3 Objectives

These motivating applications lead to the following overarching objective: Identify automation opportunities through the investigation of potential failures related to operator error. Achieving this overarching objective requires that the following six tasks be met. These include:

1. Extend function-based hierarchical models for design abstraction through integration with a process-based representation.
2. Extend outcome-driven design through integration with function-based hierarchical models.
3. Develop a formal approach to identify, propagate and rank-order failures through human-centric processes.
4. Extend function-based concept generation for the identification of solution principles of human-centric processes.
5. Develop a framework for the identification of automation opportunities in human-centric processes.
6. Demonstrate the application of the method to conceptual and reverse engineering problems through case studies.

1.4 Scope

To identify automation opportunities from human-centric processes, it is necessary to understand the processes being automated, the needs of those customers whose actions are being assisted or replaced, and the constraints that limit the adoption of new automated products that are developed.

The problems related to the identification of automation opportunities are synonymous with the issues being addressed during the engineering design process. Through the consultation of a variety of engineering design texts (Asimow 1962; Hill 1970; Earle 1990; Dieter 1991; Lindbeck 1994; Hyman 1998; Otto and Wood 2001; Dym and Little 2004; Ulrich and Eppinger 2004; Voland 2004; Niku 2009; Ullman 2010), four broad activities of engineering design may be identified; these are illustrated in Figure 1.1.

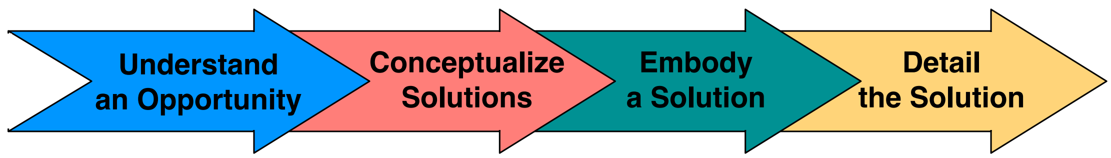


Figure 1.1 The four phases in the engineering design process.

The engineering design process can be thought of as a problem solving process. Engineering design begins with learning. The designer seeks to understand societal needs, market opportunities, and technical limitations. Through a variety of techniques the designer tries to understand what attributes a solution must have to be technically feasible, socially acceptable, environmentally conscious and economically viable. At the end of this phase of the design process, designers should understand the problem to be solved and any constraints on that solution. Next, designers explore potential solutions. They seek to investigate as wide a variety of solutions as possible. Designers draw on their creativity, intuition, and knowledge to respond to the needs and constraints embodied in the problem. Design teams explore solutions in many ways, from simple sketches to rudimentary physical prototypes. When possible solutions have been adequately explored, designers choose a concept to develop into a final realizable product. Designers choose which concept to pursue through a variety of methods from simple decision matrices to prototyping and testing of detailed concepts. As the concept begins to take form, the designers move into the embodiment phase of the engineering design process. During embodiment, designers investigate how the different components may be constructed into a complete design solution. As the solution takes shape, back-of-the-envelope calculations and rough CAD layouts check initial feasibility. But as design plans solidify, these are replaced by traditional engineering analysis and detailed CAD drawings that can be used for production. This process ends when the concept is fully converted into a realizable product.

The engineering design process may be used as a framework to scope the identification of automation opportunities. To fully understand the tasks and activities ripe for automation, it is first necessary to understand the needs of the customer. Needs must be collected from the customer and translated to the engineering domain. Once in the engineering domain, they can be explored for potential to automate; these activities all fall within the first phase of the engineering design process, *Understanding an Opportunity.* Opportunities for automation may then be explored for potential solutions. This exploration of potential solutions falls within the second phase of the engineering design process, *Conceptualize Solutions.*

This research establishes a framework that may be applied during these two early phases of the design process that often are loosely termed conceptual design. The goal of the framework is to create the underpinnings for a family of methodologies. In this dissertation, a methodology to identify opportunities for automation based on error prone, human-centric tasks is mapped to the underlying framework.

1.5 Chapter Subject Matter in Brief

Each of the chapters in this text falls within this engineering design scope covering the first two phases of the engineering design process. The goal through each of the chapters is to provide the reader with the necessary background and tools to understand and apply the methods presented in each of the following chapters either independently or together. Examples through the chapters apply the tools to different types of design problems to demonstrate the versatility of the methods. Specifically, each of the chapters discuss the following subjects.

Chapter 2 serves to further introduce the reader to the idea and domain of product design. Four distinct, yet complementary, product design approaches are presented and discussed in this chapter. Each approach focuses more completely on different phases of the product design process. A general comparison illustrates the similarities and differences between the different product design approaches.

Chapter 3 introduces formal methodologies for the generation of both functional and process models. This chapter begins broadly with an overview of abstractions in design. Functional modeling is presented as an approach to describe *what* a product must do, and process modeling is presented as an approach to describe how a customer will interact with a product. The Functional Basis (Hirtz et al. 2002) lexicon is discussed, and research into functional modeling grammars based on the Functional Basis is explored.

Chapter 4 explores the integration of process and function into a single modeling methodology. This chapter explores the relationship between function and process, and how they may be used to scope a design problem starting with customer needs. The method is demonstrated with a new design that was originally used as a student design project at the Missouri University of Science and Technology. This example illustrates how the different pieces of process and functional models integrate to form a single model structure.

Chapter 5 explores the use of customer inputs defined by the Outcome-driven Method (Ulwick 2002; Ulwick 2005) to drive the generation of functional and process models. The Outcome-driven Method, initially described in Chapter 2, provides customer inputs derived from the reasons customers purchase products. Mappings between customer inputs, functions, processes and flows are demonstrated through an existing design that was recognized as an innovative product. Demonstrating the mappings on an existing product demonstrates how this method may be applied in a redesign scenario.

Chapter 6 explores failure propagation and impact factors to identify and propagate failures through human-centric processes. The method developed in this chapter is applied to a system analysis problem to demonstrate how to study a system for potential failure points. Qualitative and quantitative impact factors are explored to sort and rank order failures based on their potential to propagate over a region of interest in the human-centric process.

Chapter 7 investigates using process models to develop functional models of human-centric actions. Process-based abstractions are assimilated with the functional representations of the tools used during the process. Empirical guidelines and a methodology are presented and are applied to a product design example. The resulting conceptual functional model is compared to a functional model generated via reverse engineering to justify the approach.

Chapter 8 develops the framework required for the identification of automation opportunities. This framework is used to formulate an overarching methodology that can be followed through the first two phases of the engineering design process to arrive at a set of concepts to solve the identified automation opportunities. Function-based and TRIZ-based concept generation are discussed as tools to assist with identifying automation solutions.

Chapter 9 applies the methodology presented in Chapter 8 to a design case study. The case study follows the first two phases of the engineering process by identifying outcomes from customer needs, mapping functionality to outcomes, propagating failures to identify automation opportunities and performing concept generation on the automation opportunities. A discussion of conclusions, broader impact and future work follow in *Chapter 10*.

CHAPTER 2 Background on Design Approaches

Amongst researchers in the area of engineering design, there are a large variety of ideas as to what activities constitute the engineering design process. Significant effort has been devoted to studying and algorithmically describing the engineering design process, but even with all this effort there is much debate as to the specific steps and boundaries. To meet the demands of each of different types of design problems, many researchers have proposed different ideas with their own combination of tools and methods that help to solve particular parts of the design process. Figure 2.1 summarizes some of these activities as a graphic where activities defined by different design texts are superimposed on four stages of the engineering design process.

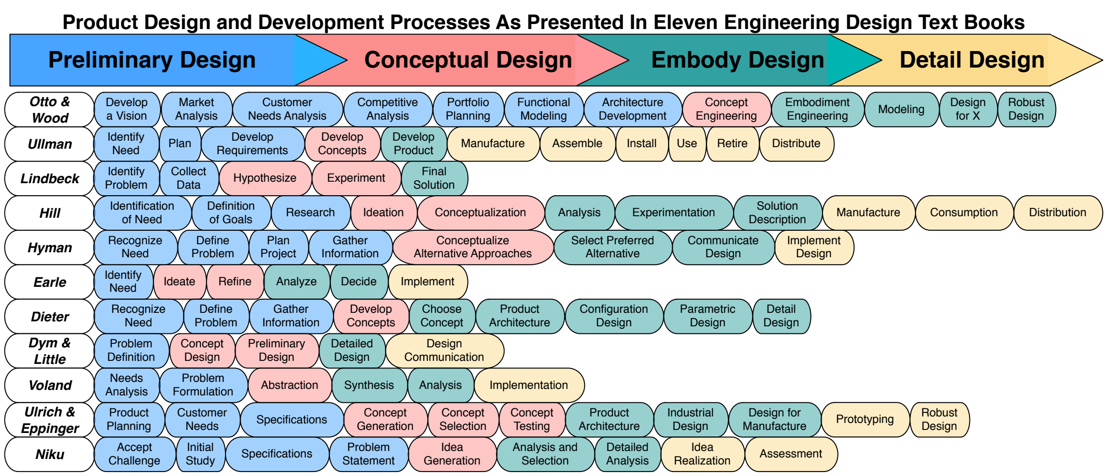


Figure 2.1 A summarization of the activities constituting the engineering design process (Hill 1970; Earle 1990; Dieter 1991; Lindbeck 1994; Hyman 1998; Otto and Wood 2001; Dym and Little 2004; Ulrich and Eppinger 2004; Voland 2004; Niku 2009; Ullman 2010).

While there are many differences between the specific tools and methods proposed, there seems to be one thing in which all of these approaches readily agree. Engineering design begins with a set of objectives or problems to solve and concludes with a solution for those objectives or problems.

In the following subsections four fundamental approaches are explored more completely to give insight into the differences and similarities between alternative techniques to solve design problems. Each approach chosen enjoys popularity within its own community, though some are more niche than widespread. Each places emphasis on different aspects of the engineering design process and proposes unique tools and methods based on their emphasis. The approaches selected include: Pahl and Beitz's Engineering Design: A Systematic Approach, Suh's Axiomatic Design, Altshuller's Theory of Inventive Problem Solving (TRIZ), and Ulwick's Outcome-driven Method. Following discussion of each approach, the differences and complementary aspects of each are explored and mapped back to a common engineering design process. The goal of this chapter is to illustrate how different design methods complimentarily work toward a common goal, while approaching design from differing perspectives.

2.1 Pahl and Beitz, *Engineering Design: A Systematic Approach*

Pahl and Beitz, in *Engineering Design*, advocate a flexible yet ordered and optimized approach to engineering design (Pahl et al. 2007). This approach, illustrated in Figure 2.2, prescribes the designer to follow systematic procedures as he or she works through a four phase design process.

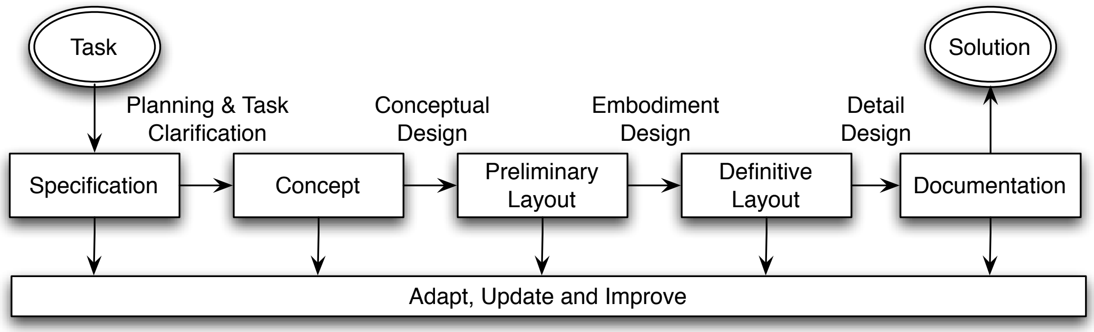


Figure 2.2 Steps comprising the systematic design process (Adapted from (Pahl et al. 2007)).

During phase one, planning and task clarification, Pahl and Beitz advocate the development of a requirements list, such as illustrated by Figure 2.3, to facilitate the collection and refinement of the customer needs for a design (Pahl et al. 2007). The process of developing a requirements list is formalized by a four step methodology where first the requirements are identified. Second the requirements are arranged in a clear order to define the main objective of the design process and split into meaningful subdivisions. Once a draft of the requirements list is developed, Pahl and Beitz, in step three, advocate using a standard template and circulating the list among interested parties for further clarifications. Any amendments to be made are incorporated during step four.

User		**Requirements List for Product**	**Identification # Date, Page**
Changes	***D/W***	***Requirements***	***Responsible***
Date of Change	**D**emand or **W**ish	Quantitative and Qualitative Date for Properties for a Product	Responsible Body

Figure 2.3 Requirements list template (Adapted from (Pahl et al. 2007)).

It is during phase two, conceptual design, where the requirements list is used to develop the specifics of the principle solution for the design. Pahl and Beitz, again, prescribe a series of steps to guide the designer though the development process (Pahl et al. 2007). First, the main design issues are identified by identifying the demands from the wishes in the requirements list to focus the direction of the design activities; then Pahl and Beitz advocate the consideration of function. The overall function (or transformation of input flows into desired output flows) of the design is identified followed by its decomposition into sub-functions of lower complexity. The resulting functional decomposition for the design, illustrated by Figure 2.4, is used to drive design conceptualization. Solution principles are identified to fulfill each of the sub-functions by either conventional methods such as literature reviews and engineering analysis, intuitive methods such as brainstorming or discursive methods such as catalogue-based searches. Working principles for each sub-function are combined in step five to develop possible design variants, and suitable combinations are selected by means of a weighted requirements-based selection criteria in step six. Often a Morphological Matrix as proposed by Zwicky (Zwicky 1969) is used to combine and analyze suitable combinations of design principles. Then in step seven, design variants are firmed-up through back-of-the-envelope calculations and models. The goal is to collect information about each design variant so that the design team can make an educated decision on engineering feasibility as well as the ability to meet the customer needs, requirements and specifications. This knowledge is used to select a final solution variant.

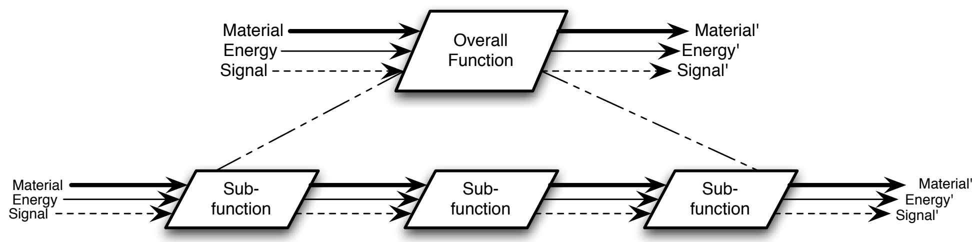


Figure 2.4 Example functional decomposition (Adapted from (Pahl et al. 2007)).

During embodiment of the design, the concept is transformed into a technical product by considering the necessary technical and economic criteria to bring the product to production. The process of embodying a design often requires a number of iterations considering alternative layouts, design forms and production processes before a definitive layout is finalized. A number of checks should be considered. Checks include: Functionality, Durability, Producibility, Assembly, Operability and Cost. Pahl and Beitz again provide a series of steps for the designer to follow (Pahl et al. 2007). The first series of steps lead the designer to a preliminary design check point and includes identifying specific embodiment requirements, producing scale drawings of spacial components, developing multiple alternative preliminary layouts and form designs, selecting suitable layouts, identifying and finding solutions to newly identified auxiliary functions, developing detailed layouts including compatibility with auxiliary solution principles, and evaluation of the preliminary layout to the requirements. To move the preliminary layout to the definitive layout, Pahl and Beitz specify procedures for optimizing and completing form designs, advocate checking for errors and other errata, and preparation of preliminary parts lists and production documentation.

And finally, during the fourth and final phase, the detailed design of the product is finalized with the formulation of the instructions required for manufacture and distribution. Documents should contain information on the final form, dimensions, materials, components, costs, assembly, et cetera (Pahl et al. 2007). Once this finalization is complete, Pahl and Beitz prescribe the integration of all component drawings into complete overall layout drawings. These are to be a part of the complete product drawings along with production, assembly, transport and operating instructions.

2.2 Suh, *Axiomatic Design*

In, *Principles of Design*, Suh argues that the process of design requires a scientific approach such that it may be taught and employed systematically (Suh 1990). With the axiomatic approach to design, Suh seeks to move the perception of design from an art—driven only by creativity and experience—to a science—driven by the scientific laws, termed axioms. Two axioms, defined as self-evident, fundamental truths similar to laws of science, are identified: the **Independence Axiom** and the **Information Axiom,** which are to be applied by a designer as he or she moves from one design domain to another to ensure the development of a *good* design.

In *Axiomatic Design: Advances and Applications*, Suh identifies four domains: the customer domain, the functional domain, the physical domain and the process domain (Suh 2001). The customer domain, shorthanded CA, contains the customer or societal needs identified at the outset of the design process. The functional domain, contains the functional requirements (FR) for a design. FRs state *what* a design must do and are solution independent. The physical domain is defined by the design parameters (DP), which state *how* the FRs will be achieved. The process domain contains the process information

(PV) for how the product will be created. During the design process, the design team maps between each of these design domains as illustrated in Figure 2.5. During this mapping, first the design team tries to maintain the independence axiom (e.g., a single FR maps to a single DP). When multiple designs meet the independence axiom, the information axiom is utilized to determine the *best* design by identifying the design where the least amount of information is required to successfully achieve the domain mappings.

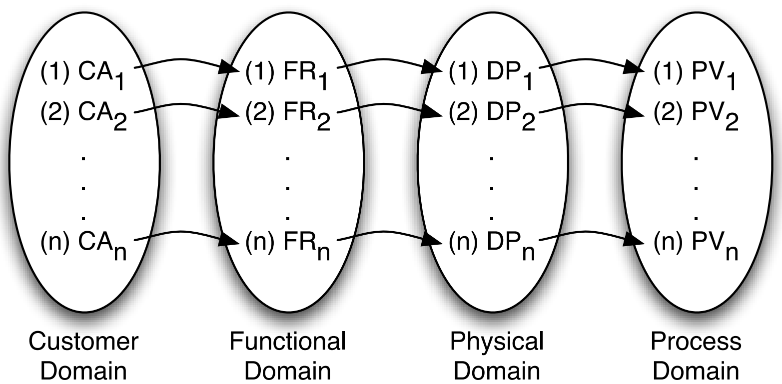


Figure 2.5 Mapping between the customer, functional, physical and process design domains (Adapted from (Suh 2001)).

Mathematical approaches are presented to identify independence and to quantify information content in a design. To calculate independence, Suh provides the design equation (2.1) (Suh 1990).

$$\{FR\} = [A]\{DP\} \quad (2.1)$$

The design equation (2.1), uses [A] as the design matrix to characterize the design (Suh 1990; Suh 2001). For linear designs, the values of the design matrix take the form of constants relating mapped design domains (e.g. DP mapped to the FR); for nonlinear designs, functions relate mapped design domains.

Designs may be either uncoupled (i.e., independent), coupled or decoupled. These domains can best be illustrated by the matrices shown in Figure 2.6 based on Equation 2.1. In Figure 2.6, the mappings between the functional domain and the physical domain are provided as examples of coupled, uncoupled and decoupled designs. In an uncoupled design, changes to one FR only affects the DP to which it is linked. However, in a coupled design, changes to one FR can affect every other DP in the design. When a design is decoupled, changes can be made to an FR without affecting other DPs, but in order to maintain this independence the changes must follow the order of the mappings as they are represented in the design matrix.

Uncoupled

$$[\mathbf{A}] = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & A_{mn} \end{bmatrix}$$

$$\begin{aligned} FR_1 &= A_{11}DP_1 \\ FR_2 &= A_{22}DP_2 \\ &\vdots \\ FR_n &= A_{mn}DP_m \end{aligned}$$

Coupled

$$[\mathbf{A}] = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

$$\begin{aligned} FR_1 &= A_{11}DP_1 + A_{12}DP_2 + \ldots + A_{1n}DP_m \\ FR_2 &= A_{21}DP_1 + A_{22}DP_2 + \ldots + A_{2n}DP_m \\ &\vdots \\ FR_n &= A_{m1}DP_1 + A_{m2}DP_2 + \ldots + A_{mn}DP_m \end{aligned}$$

Decoupled

$$[\mathbf{A}] = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

$$\begin{aligned} FR_1 &= A_{11}DP_1 \\ FR_2 &= A_{21}DP_1 + A_{22}DP_2 \\ &\vdots \\ FR_n &= A_{m1}DP_1 + A_{m2}DP_2 + \ldots + A_{mn}DP_m \end{aligned}$$

Figure 2.6 Matrix representations of uncoupled, coupled and decoupled designs (Adapted from (Suh 1990)).

The second axiom states that information must be minimized. Information is defined as the probability of mappings successfully being met when realized in the final design (e.g., a DP successfully meeting an FR). Information may be quantified in terms of the range and tolerances for parameters. Suh provides Equation (2.2) for this quantification (Suh 1990).

$$I = \log\left({}^{range}\!/_{tolerance} \right) \tag{2.2}$$

Again, if we consider the mappings between FR and DP as an example, DPs with the least information content are the most desirable since they are the most likely to be manufacturable.

Design progresses by mapping between domains iteratively. First an FR is defined. Then the FR is mapped to a DP. The FRs are alternatively decomposed hierarchically and mapped to DPs. This, in effect, zig-zags between the design domains to map the customer domain to the functional domain, the functional domain to the physical domain, and the physical domain to the process domain. Independence and information axioms should be maintained through all mappings, and once mappings are made, independence is verified to identify *good* designs. If more than one design is uncoupled, then the *best* design is selected by applying the information axiom. This process is repeated to map between all four domains throughout the design process.

2.3 Altshuller, *Theory of Inventive Problem Solving (TRIZ)*

Altshuller proposes that the ideas of creativity and innovation may be controllable as scientific processes by considering specific problem pieces instead of the problem as a whole (Altshuller 1995). Altshuller proposes five levels of problems that increase in complexity. The first level are those with which the solution to the problem does not change the design artifact. In the second level, the solution changes the design artifact, but the changes are not substantial. For the third level, the essence of the design artifact becomes different through changes in the interactions between its components. In the fourth level, a solution to the problem would result in a totally changed design artifact, and in the fifth level, the solution requires a completely new technical system for the changed design artifact. Each of these levels result in an increase of potential solutions, and for an engineer to solve the problem, each solution must be investigated. Consider that from the first to second level of problem, the number of potential solutions goes from a few to a dozen, but for the fifth level, the number of potential solutions increases to hundreds of thousands to millions. Plus, when an engineer can only practically deal with 50-70 solutions and each potential solution must be investigated, the task of solving a fourth or fifth level of problem as a whole becomes daunting at best.

Key to solving the higher-level problems is the identification of and resolution of contradictions. There are three types of contradictions: Administrative (AC), Technical (TC) and Physical (PC). ACs are high-level problems that occur when something needs to be done, but how to do what needs to be done is unknown. TCs occur when a change to one part renders another part unusable, and PCs occur when mutually opposing demands are placed on a single element in a system. PCs are the most difficult contradiction to overcome and tend to be associated with the fifth level problems, while ACs are the least difficult contradiction and tend to be associated with the first level problems (Altshuller 1995). Altshuller proposes that the route to finding a solution should be different depending on the type of contradiction. For example, Altshuller proposes 40 fundamental principles to solve TCs. The 40 Principles are the direct result of a study of 200,000+ patents that revealed about

1,500 different TCs. Each of these 1,500 TCs were solved through the application of fundamental principles. These principles, in practice, are meant to provide guidance on how to remove a TC from a technical system (Altshuller 2005).

To further illustrate this concept of resolving a contradiction, consider a classic TRIZ example found in (Altshuller 2005). Cargo vessels must transport cargo in the winter across water ways frequently covered by ice. Ice breakers typically open the path before traditional cargo vessels pass, but ice breakers with the most powerful engines still operate too slowly. Alternative transportation methods are not available. To solve this problem, the technical contradictions are first identified. For this example, two may be identified, *speed* versus *power* and *productivity* versus *power.* To assist with identification of these contradictions, 39 characteristics of technical systems are provided in TRIZ. Once the TCs are identified, a contradiction matrix is used to link each TC to the Principles known to provide solutions. Four Principles are identified: Principle #19, *Periodic Action*; Principle #35, *Transformation of Properties*; Principle #2, *Extraction*; and Principle #10, *Prior Action*. From these principles, a proposed solution suggests changing the profile of the ice breaker to decrease the surface area contacting the ice, while also increasing the size of the ice breaker so that it can also act as the cargo ship (Altshuller 2005).

This approach, however, only works for solving TCs. To solve the more difficult PCs a reduction of complexity must occur. To reduce problems from the fifth level (PC) to the first level (AC), the Algorithm to Solve an Inventive Problem (ARIZ or ASIP) is proposed (Altshuller 1995; Altshuller 2005). ARIZ is the primary analytical tool for TRIZ. ARIZ, shown graphically in Figure 2.7, provides a sequence of nine steps to reduce problem type and solve a problem.

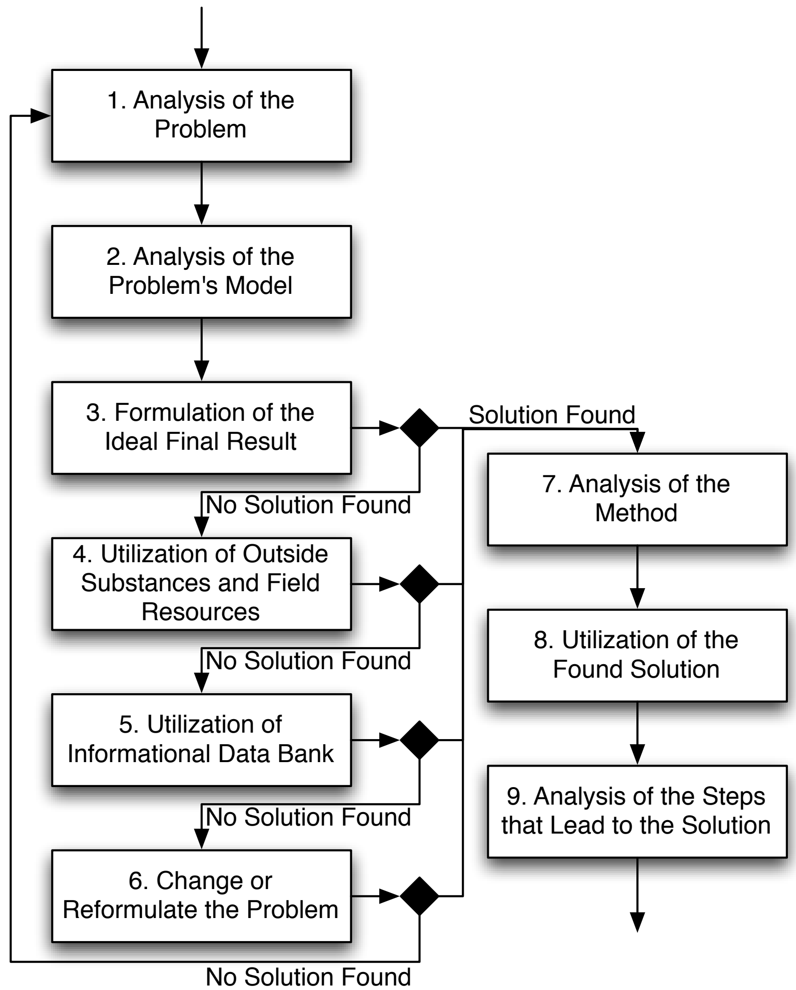


Figure 2.7 Algorithm to Solve an Inventive Problem (ARIZ) (Adapted from (Altshuller 1995)).

During step one, Analysis of the Problem, problem statements are to be translated from being broadly defined to very specific mini-problems, and the conflicting situation—termed **contradiction**—is identified. Once the problem is identified, a Substance-Field (S-Field) model is created of the Operating Zone in step two. The S-Field model, based on the interactions shown in the

Key in Figure 2.8, provides a succinct representation of how substances and fields interact. Substances in the S-Field are any materials involved with the contradiction, while the field is any source of energy. For example, in Figure 2.8, the Field, F, interacts negatively with S_3. The result of this interaction is a directional action on S_2 and a negative action on S_1. The 72 Standards are used to assist with modifying the S-Field to rectify problems identified during analysis (Altshuller 2005).

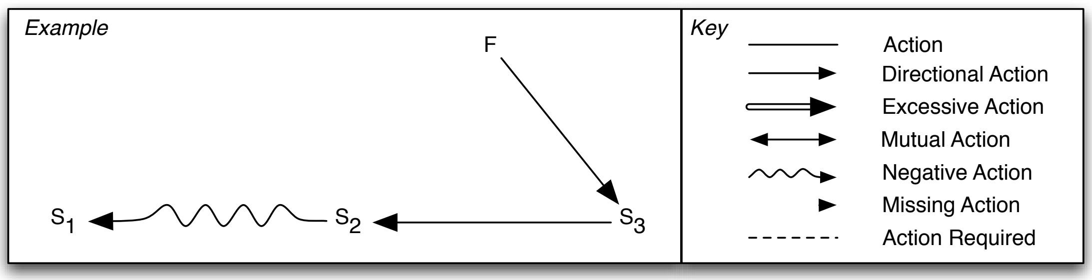


Figure 2.8 Example S-Field and Key for translating S-Field interactions (Key adapted from (Altshuller 1995)).

After S-Fields, the third step guides the formulation of the Ideal Final Result or IFR. The IFR is an *idealized* final result from the solution to a conflict and is meant to provide a goal toward which a problem solving process will strive. If a solution is not found during these first three steps and the problem is still unclear, then Steps 4-6 should be consecutively applied stopping with the step where a solution is found. Step four uses the "Little Man" model as an imaginative tool where the problem is considered in context of little men performing the operations of the system (Altshuller 1995; Altshuller 2005). Step five suggests using one of Altshuller's 72 Standards. Step six suggests to reformulate the contradiction from the original super-system; this is the problem solving step and should be repeated until a solution is found. Once the

PC is removed, the method that was used to remove the PC is analyzed to check the quality of the solution. The solution is implemented in Step 8, and in Step 9, any deviations from the ARIZ are noted for future problems.

2.4 Ulwick, *Outcome-driven Method*

In *What Customers Want: Using Outcome-Driven Innovation to Create Breakthrough Products and Services*, Ulwick proposes a customer-based approach to design. The argument is that new products fail because designers collect and develop products from the wrong inputs (Ulwick 2005). As inputs, Ulwick proposes that designers stop using customer needs and instead identify the customers' jobs, outcomes and constraints. Ulwick argues that customer needs are unreliable because they are customer speak requirements lacking the proper structure required to drive innovative design, and their translation to proper design metrics is difficult and often ambiguous leaving gaps in the requirement data. To remove this ambiguity, Ulwick proposes identifying specific customer inputs—jobs, outcomes and constraints—which can be directly translated to requirement data removing the ambiguity in the innovative design process. These new inputs are derived from the three key tenets. The tenets state that (1) customers look to purchase new products to help complete functional tasks, and (2) will evaluate the performance of the product based on a set of metrics, which (3) enable a systematic design process (Ulwick 2005).

Following these three key tenets, Ulwick proposes an eight step methodology shown in Figure 2.9 that is comprised of two phases: (1) Identifying Opportunities and (2) Addressing Opportunities.

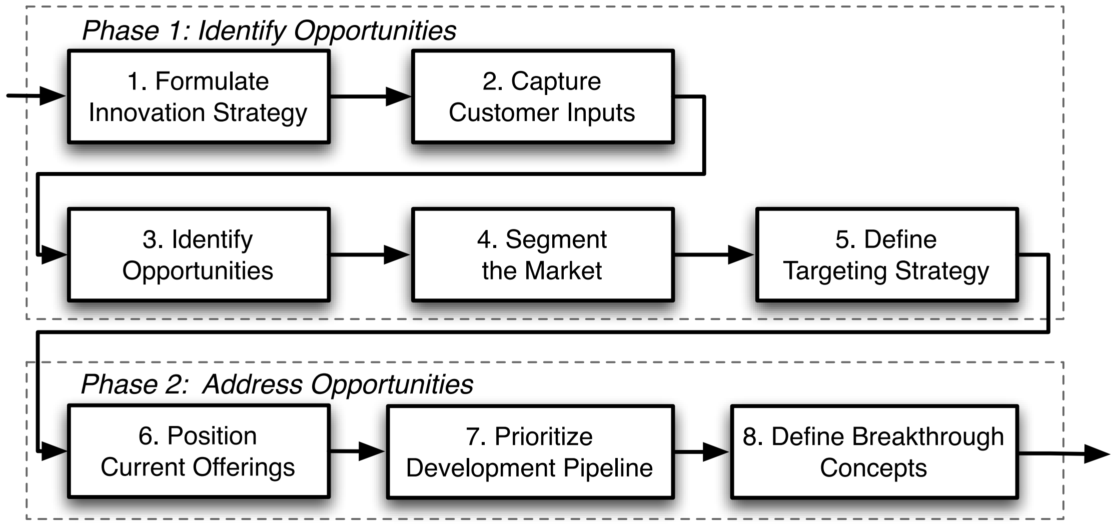


Figure 2.9 Outcome-driven method (Adapted from (Ulwick 2005)).

Following Ulwick's method, the design team first focuses on what types of opportunities are available, what customers (i.e., end user, supplier) to address, and what growth options are of interest. Once a direction is chosen, the customer inputs are identified. These customer inputs include the customer's job, outcomes (metrics used to measure success), and constraints (limitations on the design). The jobs and outcomes are sorted to identify which present opportunities for new product development; for this, Ulwick proposes using the opportunity score, provided as (2.3) (Ulwick 2005).

$$O = I + MAX(I - S, 0) \tag{2.3}$$

The opportunity score, O, is calculated following (2.3) where the Importance value, I, and the Satisfaction value, S, are determined from customer surveys. Both I and S are collected on a 5-point Likert scale and doubled for calculations. Opportunity scores over 15 represent the highest opportunity outcomes. Scores between 12 and 15 represent opportunities for design improvement. Scores between 10 and 12 should be considered, while scores less than 10 should be disregarded (Ulwick 2005). Once opportunities are identified, the market should be segmented based on desirable and undesirable outcomes. A strategy is now formulated for how the design team will address the identified opportunity.

With a strategy in hand, the design team moves on to the second phase, Address Opportunities (Ulwick 2005). Here, the design team has three final steps. They are first advised to look at the company's current offerings to identify products or services that meet identified opportunities. Then, they are advised to prioritize those products in the development pipeline with potential to meet the identified opportunities, and finally, they develop a new "breakthrough" product utilizing focused brainstorming. Those products which best address the customer's outcomes should be selected for further development.

2.5 Comparison of the Alternative Design Approaches

Each of these four approaches, while being systematic in their pursuit of solutions to problems, addresses the product design process from a different perspective. These different perspectives influence the specific tools and methods proposed by each of the approaches.

2.5.1 A Comparison of Perspectives

The systematic approach to design developed by Pahl and Beitz seeks to develop creative workable solutions. They are less concerned with finding the best solution, than with generating many good solutions from which one can be selected. Little focus is placed on identifying customer needs beyond identifying obvious market opportunities. Likewise little effort is devoted to integrating manufacture into the design beyond offering some basic heuristics for fits and tolerances. The only focus given to manufacturing is in a section on design for production where the focus is on the specific materials used, the shape (for manufacturability) of the final parts, the production considerations for various architecture types, and considerations specific to individual manufacturing processes. A section on design for assembly follows manufacturing and focuses on the application of standard parts, design layouts, and standard manufacturing operations, et cetera (Pahl et al. 2007).

Suh is primarily concerned with enabling robust and concurrent design through mapping customer needs to functionality and function to manufacturing processes. A principal aim is to treat design as science; there is one best solution according to the stated axioms. Customer needs are important, but gathering them is not an explicit feature of the method. Conceptual design is considered concurrently with embodiment to promote manufacturable solutions. The creation of robust solutions places manufacture as a vital piece of the design process by considering tolerances via the Information Axiom throughout the CA, FR, DP and PV mappings (Suh 1990; Suh 2001).

Altshuller seeks to facilitate novel, patentable solutions. His method is not concerned with enabling routine design tasks and is focused on situations where technical contradictions demand inventive solutions. Customer needs gathering is not a facet of TRIZ; a thorough understanding of the problem is assumed. Nor is manufacturability of the solution a concern; it is, however, possible to resolve manufacturing contradictions following TRIZ. Once TRIZ is followed, it is possible that the resolution of the technical contradiction may not be physically possible. Altshuller's method thrives on solving difficult technical problems and is generally applicable throughout the design process (Altshuller 1995; Altshuller 2005).

The focus of Ulwick's approach is on collection of the proper customer inputs to allow for innovative design, while also facilitating the ability of the customer to complete jobs. The inputs to the process define the job to be completed by the customer, the metrics that the customer will use to measure their satisfaction and the limiters to adoption by the customer of the new product. These inputs lead to product opportunities. While the method outlined by Ulwick consists of two phases—*identifying opportunities* and *addressing opportunities*—the primary focus falls in the steps that make up the initial phase. To address opportunities, Ulwick presumes that the business unit designing the products may either have a product to be re-marketed or in the development pipeline. For those that do not, brainstorming is suggested as a potential (or perhaps even primary) approach for concept/product development. The following steps that generally make up the engineering design process—embody design and detail design—are not addressed in the Outcome-driven Method (Ulwick 2002; Ulwick 2005).

The different perspectives taken by each approach allows each to focus on different parts of the design process more fully. Each approach excels in the areas where its tools focus, while other methods are less developed. This does not necessarily mean that one method is better than another. In fact, it means that the different methods complement each other rather than compete with each other. Each method is well suited to solve a particular type of problem and may be used in conjunction with each other throughout the design process.

2.5.2 Mapping to the Engineering Design Process

Stemming from their difference in perspective, each approach addresses different key aspects of the engineering design process. In Figure 2.10 each approach is mapped to a common engineering design process to illustrate these differences.

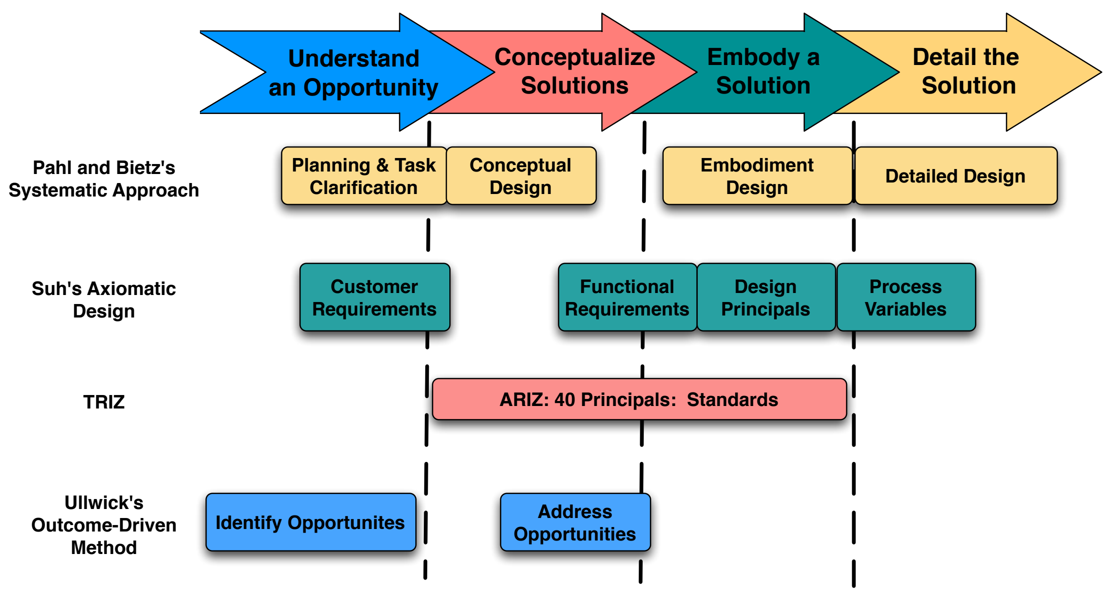


Figure 2.10 Mapping each of the four design approaches to a common engineering design process (Suh 1990; Altshuller 1995; Suh 2001; Ulwick 2005; Pahl et al. 2007).

Pahl and Beitz with their creativity focus have specific tools to address the conceptual and the detailed design portions of the process. Functional models are used to systematically capture what a product must do. Morphological matrices are used to map solutions to functions. Objective trees are used to evaluate customer needs. Suh, however, initially focused Axiomatic

Design on the later phases mapping FR to DP to PV. This mapping is iterative going between the domains, but does generally begin after customer needs gathering. In newer versions, CR mapping has been included to capture more of the engineering design process. Altshuller, with ARIZ, begins with a problem and works to solve the problem. This process assumes that a problem has been found and that a need exists, and while this process may be used during any phase in the engineering design process, no explicit focus is given to the manufacturability of the final solution. Ulwick, however, focuses almost exclusively on the front end of the engineering design process to collect customer inputs. Little focus is provided on the conceptual, detail or implementation phases.

2.6 Summary

Each of these four techniques to solve design problems approach design from a different perspective providing unique tools well suited for different phases of a common engineering design process. Outcome-driven Method places focus on ensuring the proper customer inputs are available for the design process. Systematic Approach develops tools and methods to enable on a creative and innovative design process. The goal of axiomatic design is to create a scientific approach to design ensuring that the best design is found in the design process. And, in TRIZ, Altshuller creates an algorithm to facilitate the creation of innovate solutions to complex problems.

Each method is complementary and may be applied as a single contiguous method as illustrated in Figure 2.10. However, since the scope of this research is in the first two phases of the design process—understanding opportunities and conceptual design—the design approaches on which we will focus and extend include Outcome-driven Method and Systematic Design. Ulwick's Outcome-driven Method will initially be used during the understanding opportunities phase to collect customer inputs related to why customers purchase products. Then function-based design—as advocated by Pahl and Beitz—will be be used to transform these customer inputs to functionality that can be used to drive design conceptualization.

CHAPTER 3 Functional and Process Modeling

Function and process used during engineering design provide abstractions to capture transformations related to why a customer needs a product and what a product needs to do. Abstractions such as function and process are fundamental to the engineering design process. Without appropriate abstractions, many complex problems could not be solved (Voland 2004). An abstraction, by definition, allows a problem to be extracted from its physical reality in such a way as to provide a means for problem solvers to solve those problems, which prior to abstraction, exist as nothing more than an idea ("Abstraction" 2001; Abstraction" 2005). Abstractions take various forms in all branches of engineering. For instance, free-body diagrams generated in mechanics are used to solve equilibrium problems by extracting a particle from its surroundings and applying known force loads. Electrical schematics provide abstractions of circuits with symbols to represent components such that a circuit may be readily understood, analyzed and constructed. Control engineers generate block diagrams linking dynamical attributes of a system to gain understanding of overall system dynamics. Computer-Aided Design (CAD) provides numerous potential abstractions for engineers and designers to convey information to managers, customers and marketing. The abstractions developed in solid modeling utilizing CAD technology also provide a means to package complex systems, analyze stresses and isolate failures. Broadly, each of these abstractions may be considered a model, where a model, as defined in Webster's Encyclopedic Unabridged Dictionary, is "a simplified representation of a system or phenomenon, as in the science or economics, with any hypotheses required to describe the system or explain the phenomenon, often mathematically" ("Model" 2001).

3.1 Functional Modeling

In engineering design, functional abstractions (often presented as a functional model or their analog function structures) are often considered a key part of the engineering design process, and their use is advocated in numerous pieces of engineering design literature (Miles 1961; Roth 1981; Hundal 1990; Suh 1990; Dieter 1991; Cutherell 1996; Cross 2000; Otto and Wood 2001; Suh 2001; Ulrich and Eppinger 2004; Pahl et al. 2007; Ullman 2010). Function, when considered during preliminary design, provides flexible models for problem abstraction which can help answer numerous design questions with a focus "on what has to be achieved by a new concept or redesign and not how it is to be achieved" (Otto and Wood 2001). An abstraction, such as a functional model, based on what a product must do instead of how it will be done provides the benefits of an explicit relationship to customer needs, comprehensive understanding of the design problem, enhanced creativity, innovative concept generation, and systematic organization of both design problems and the design team (Otto and Wood 2001; Ullman 2010). Terminology related to functional modeling may be formally defined as:

- **Functional Modeling:** The overall approach to modeling what a product must do in terms of elementary operations such that the product may achieve an overall objective (Stone et al. 2000).
- **Flow:** A material, energy or signal, which interacts with the product; flows are expressed as nouns (Stone et al. 2000).

- **Function:** A description of an operation, expressed as the active verb, performed by an artifact, as a part of a larger product, to transform an input flow to a desired output flow (Stone et al. 2000). Functions are tied together via material, energy and signal flows.

3.1.1 Functional Modeling Background

Numerous parallel functional modeling techniques have been proposed and researched to aid with product design (Chandrasekaran 1994; Erden et al. 2008). Commonly, these functional modeling techniques have a representation for the functionality of the system, the structural state of the system and behavioral expectations for the system. In these approaches, behavior commonly represents the change in state or action of the physical form of the system in response to a stimuli—defined in (Eder and Hosnedl 2008). For example, in Umeda and Tomiyama's Function-Behavior-State (termed Function-Behavior-Structure in (Umeda et al. 1990)), state is the physical description of an entity in a design, behavior is the change in the state and functionality is how the behavior of the system will be realized through design (Umeda et al. 1990). Structure-Behavior-Function similarly uses structure to represent a physical description for components, function is the pre- and post- conditions for the behavior of the system, and behavior is the transition between states (Goel and Chandrasekaran 1992; Goel et al. 2009).

Welch and Dixon, however, define behavior as how the system will meet the required functionality. Function, in their approach, defines what a system is going to do, and the conceptual design process is the transition from function to behavior to structure (Welch and Dixon 1992). Function-Behavior-

Structure, developed by Gero, similarly follows the evolution as a conceptual design moves from function variables to behavior variables to structure variables (Gero 1990). Function variables represent requirements for the design; behavior variables represent the intended or anticipated actions of the final system, and structure variables represent the physical form of the system. To capture environmental interactions, Gero's approach is expanded with Situated Function-Behavior-Structure (Gero and Kannengiesser 2002). Behavior-driven Function-Environment-Structure (B-FES) modeling framework similarly includes representation for environmental interactions of the system and propose a direct mapping from function to behavior to physical structure (Tor et al. 2002; Zhang et al. 2002).

3.1.2 Functional Modeling with the Functional Basis

Functional modeling with the Functional Basis can trace its roots back to Value Analysis with the work of Miles (Miles 1961) and Rodenacker (Rodenacker 1971). This early work in Value Analysis is expanded through the proposal of additional functions by Roth (Roth 1981), which is further formalized through Koller's proposal of twelve basic functions (Koller 1985). At a high level of abstraction, Pahl and Beitz develop a list of five generally accepted functions and three flow types (Pahl and Beitz 1984). Hundal then proposes a set of six function classes in (Hundal 1990), but excludes the flow of information, which are re-added to the structure by Little et al., with the functional basis set (Little et al. 1997). Standardized sets of function and flow terms are proposed separately by Szykman (Szykman et al. 1999) and Stone

(Stone and Wood 2000). These function and flow terms are reconciled by Hirtz et al. to form the reconciled Functional Basis (Hirtz et al. 2002).

The reconciled Functional Basis consists of terminology to describe all functions and flows for electromechanical systems (Hirtz et al. 2002). Terminology for functions and flows are comprised of three levels of detail termed *classes*: primary, secondary, and tertiary. Table 3.1 and 3.2 reproduce the primary and secondary classes of the reconciled Functional Basis.

Table 3.1 Primary and secondary flow classes of the Functional Basis (Adapted from (Hirtz et al. 2002)).

(Class) **Primary**	Material	Signal	Energy		
Secondary	Human	Status	Human	Electrical	Mechanical
	Gas	Control	Acoustic	Electromagnetic	Pneumatic
	Liquid		Biological	Hydraulic	Radioactive
	Solid		Chemical	Magnetic	Thermal
	Plasma				
	Mixture				

Table 3.2 Primary and secondary function classes of the Functional Basis (Adapted from (Hirtz et al. 2002)).

(Class) **Primary**	Branch	Channel	Connect	Control Magnitude	Convert	Provision	Signal	Support
Secondary	Separate	Import	Couple	Actuate	Convert	Store	Sense	Stabilize
	Distribute	Export	Mix	Regulate		Supply	Indicate	Secure
		Transfer		Change			Process	Position
		Guide		Stop				

To further improve functional model consistency when using the Functional Basis, effort has been taken to evolve the Functional Basis into a formal modeling language. This effort considers both the meaning of a traditional formal language and the Functional Basis as the underpinnings of a language to evolve the the Functional Basis into the Functional Basis Modeling Language (Nagel, Vucovich et al. 2007; Nagel, Vucovich et al. 2008).

Analogies are drawn between the structure of a traditional language and functional modeling with the Functional Basis. Traditionally a language for human communication consists of five parts: phonology, morphology, syntax, lexicon, and semantics; however, if the language is written as well as spoken, a sixth part, graphology is added to the language's structure (Millward 1996). The smallest meaningful units of functional modeling are function and flow terms, and as such, they are the morphemes of the language. The arrangement of function and flow terms into function-flow pairs is the morphology of functional modeling. The Functional Basis lists all the morphemes of functional modeling (functions and flows) and thus is the lexicon of functional modeling. Semantics is the study of the meanings of functions, flows, and their pairs (much of which is found in the definitions), and syntax is the arrangement of function-flow pairs into independent function chains and aggregated functional models. Graphology is the written representation of functional modeling through function blocks and flow arrows to form meaningful functional models.

Grammar is the glue employed to pull together all the different parts of a formal language and to thus provide standardized structure. In formal languages words come together into larger meaningful structures following the rules of a grammar where grammar defines language structure and consists of both a syntax and a morphology (Quirk et al. 1985; Grammar" 2005). Thus, both a clear morphology and syntax are required to guide the meaningful and consistent joining of function-flow pairs taken from the Functional Basis. Appendix A provides a working morphology for the Functional Basis, while Appendix B provides a working syntax. The morphology and syntax are termed working, because like with a traditional language, the way in which morphemes fit together must be flexible evolving to address new modeling applications not originally considered.

3.1.3 Functional Modeling Methodology

Functional modeling with the Functional Basis provides a technique to model the transformation of flows within a product via a standard lexicon. Generally, functional models consist of at least two levels. At the top level is a black box functional model describing the overall functionality of the product. At the second level is a functional (or sub-functional) model detailing functional changes on each flow through the product. These model layers may be defined as:

- **Black Box (Functional) Model:** The high-level functional model defined by a single overall operation of the product being designed.
- **Functional Model:** A structured arrangement of functional elements tied together via material, energy and signal flows describing an artifact or collection of artifacts that collectively comprise the product (Stone et al. 2000).

Generally, the following five steps are followed when generating a functional model (Stone and Wood 2000):

1. Understand the needs of the customers.
2. Generate a black box model capturing overall functionality of the product as well as all flows. This high-level functionality and flows are based on the customer needs.
3. Decompose each flow into function chains capturing the changes required to change each flow from its input to its desired output.
4. Aggregate function chains into a complete functional model.
5. Verify that the customer needs identified during Step 1 are addressed in the functional models.

Black box functional models are stand alone functional models abstracting a high-level transformation intended for the product to complete. The black box functional model is derived from customers' needs identified during Step 1. This black box functional model, shown in Figure 3.1, is labeled with the high-level transformation intended for the product to complete. The input and output flows identify all flows required for the operation of the product. Material flows are bold arrows; energy are thin arrows and signals are dashed arrows. The three types of flows are drawn entering and exiting the black box functional model as shown in Figure 3.1.

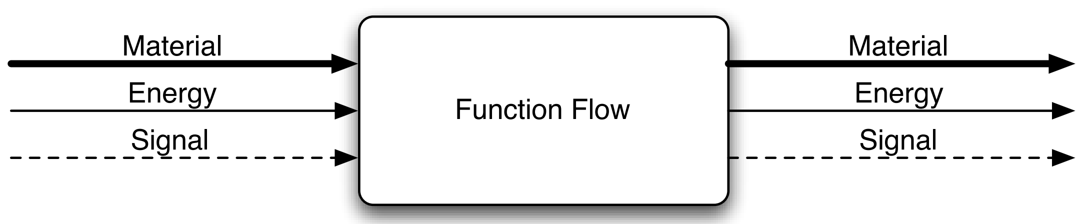


Figure 3.1 Example black box functional model.

A functional (often termed sub-functional) model decomposes the overall functional black box into specific flow transformations. These transformations define the operations required of the system such that the identified input flows do become the identified output flows through the operation of the system. The decomposition of the black box into a functional model begins by first detailing the transformations to each flow. A simple way to develop these transformations is to consider yourself as the flow; this is termed, *being the flow* (Otto and Wood 2001). This creates chains of flow transformations. These chains of flow transformations are then aggregated into a single functional model as illustrated in Figure 3.2.

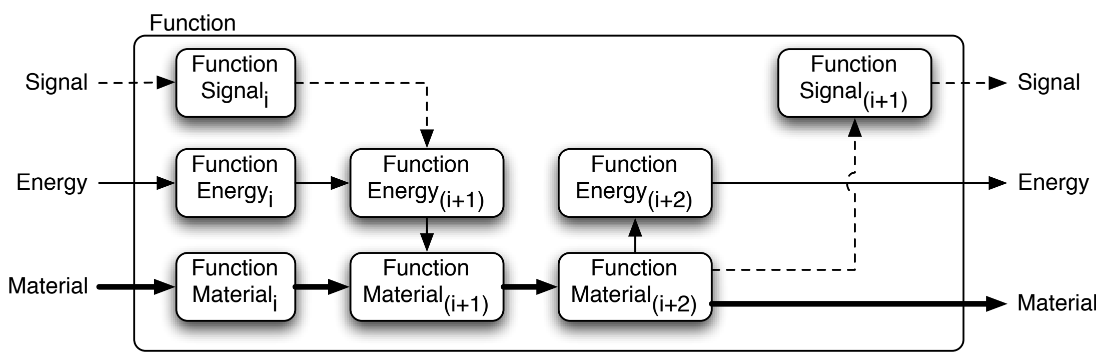


Figure 3.2 Example functional model for the black box functional model.

3.2 Process Modeling

As the functional abstractions generated during the design process grow to incorporate changes to the product, accommodating human-based operations and environmental interactions, the line between traditional functional modeling techniques and process-based project management techniques begins to blur. To fill this void, a process modeling methodology based on functional modeling with the Functional Basis is used to model configurations, events and processes. Formally, the terminology related to process modeling may be defined as:

- **Process Modeling:** The overall approach to modeling a series of customer-driven, product-based operations related through input and output flows, the product being designed, and time.
- **Configuration:** A specific discrete instance of the overall function of the product which may relate to the environment where the product is used or specific applications of the product. The configuration of a product is modeled functionally.
- **Event:** A set of configurations of a product, which pertain to changes to the operability of a product or sequencing of operations during the usage of a product.
- **Process:** The set of defined events that occur with respect to the product as a whole and aim to meet a particular goal. Processes are tied together via the product, material, energy and signal flows.

Process modeling adds breadth to formal functional modeling techniques by enabling increased fidelity of modeling abstractions generated during product design. Where functional models provide depth through hierarchical models of single individual configurations of a product, process models provide breadth by capturing customer-product interactions through the unique events and configurations of the product. Process models share similarity with project planning methods, activity diagrams and user-centered design. The following background section discusses these modeling activities.

3.2.1 Process Modeling Background

To model processes, traditional approaches such as Program Evaluation and Review Technique (PERT) (Malcolm et al. 1959) and Critical-Path Method (CPM) (Kelley and Walker 1959) and Activity Diagrams (Otto and Wood 2001) are often the most recognized. These stem from multiple domains, but may each be used to abstract a process. PERT and CPM for example stem from project planning. A PERT chart, illustrated in Figure 3.3 (a), is typically used to model events during the design process. The Critical-Path is then the single longest chain of sequential events as represented in the PERT chart that must be completed during a project (Kelley and Walker 1959; Malcolm et al. 1959; Ulrich and Eppinger 2004).

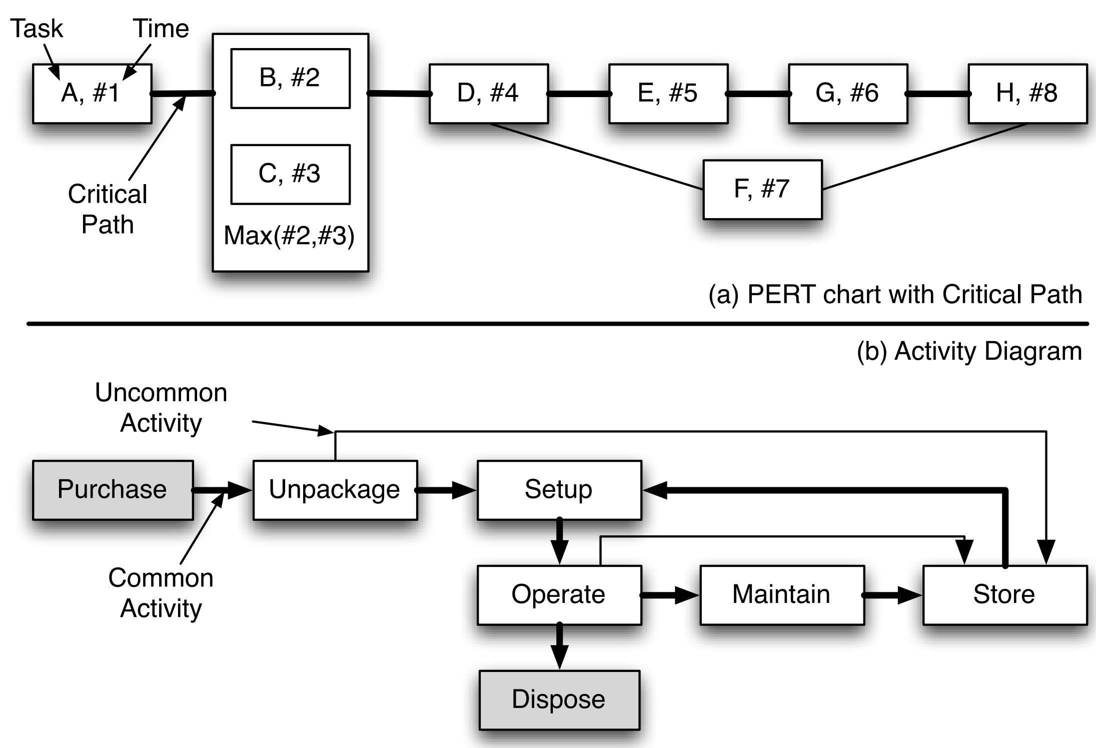


Figure 3.3 Example (a) PERT chart with Critical Path marked in bold (Adapted from (Ulrich and Eppinger 2004)) and (b) activity diagram (Adapted from (Otto and Wood 2001)).

Activity diagrams, while similar in their structure, stem from the product design domain. Activity diagrams, illustrated in Figure 3.3 (b), are a form of block diagram that allow a designer to capture the customer's distinct product uses and use patterns over the life of a product (Otto and Wood 2001). Each of the blocks in an activity diagram are similar to the events defined above; however, they are nonhierarchical and they do not integrate with other model representations.

Other approaches such as Workflow Planning (Ju 2001), Workflow Activity Models (WAMO) (Eder and Liebhard 1995) and Process Flow Diagrams (Ulrich and Eppinger 2004) add features such as capturing critical review points and milestones, flow of project resources, and basic scheduling information. Workflow planning, as proposed by Ju, defines each process as a single workflow comprised of one or more sub-processes (Ju 2001). Managers are free to coordinate and make decisions on the sub-processes within each workflow. WAMO is based on the general idea that any business process can be broken down into smaller working units or workflows, which can again be broken down into smaller sub-processes (Eder and Liebhard 1995). A key difference, however, is that Eder and Liebhard propose visualizing each workflow as an activity tree such that a hierarchy is created where activities contain sub-activities. Process flow diagrams, like the aforementioned project planning techniques, have been developed to model workflows. Process flow diagrams, instead, focus on the workflow of the design team. A block diagram details the teams' actions during the design process, and can include critical review points, program approval goals and other design milestones (Ulrich and Eppinger 2004). A similar approach, proposed by Andersson et al., more rigorously models the design process providing a framework for the application of sensitivity analysis within the design process modeling structure. Model elements capture tasks and their characteristics as well as design reviews and their probability for success (Andersson et al. 1998).

Similarly, in the engineering design domain, a Design Structure Matrix (DSM), while traditionally being used to represent forward and backward relationships between product components, may also be used to represent sequential, parallel and coupling of tasks and resources required for a process (Ulrich and Eppinger 2004). The DSM illustrated in Figure 3.4 may then be used for evaluation of process execution strategies using optimizations designed to clump tasks based on their dependences (Cho and Eppinger 2001).

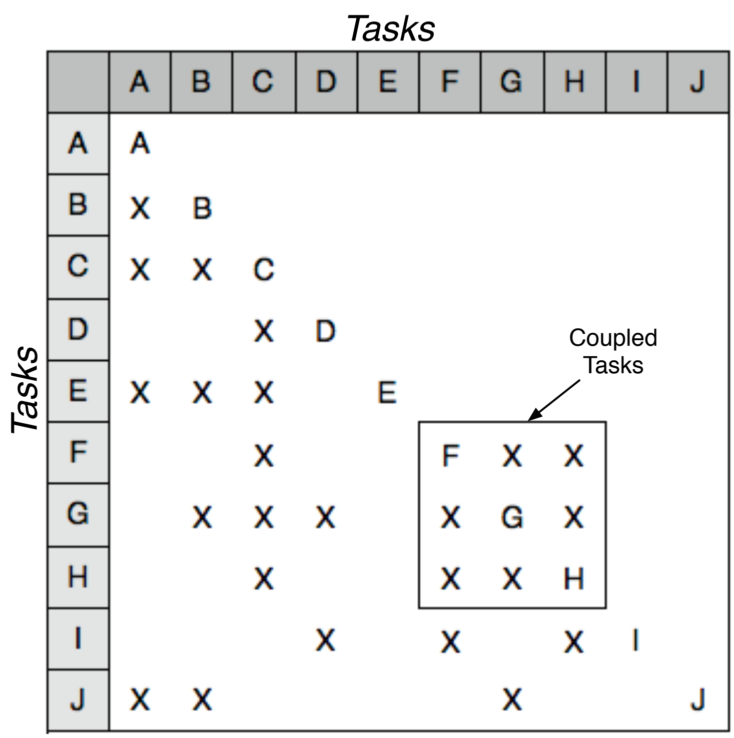


Figure 3.4 Example Design Structure Matrix (Adapted from (Ulrich and Eppinger 2004)).

A difficulty, however, with the application of a DSM for design process modeling is the elicitation of thorough and accurate process information; to overcome this difficulty, Wallace et al. propose an open-marketplace approach to collect process information and auto-propagate a DSM (Cho and Eppinger 2001). The auto-propagated DSM can subsequently be optimized following DSM-based approaches. A DSM has also been utilized as a tool to tie the design process to the evolution of a design (Fagerström and Nilsson 2003) where function-means structures jointly model evolving product functionality and solution strategies, and the product development process is modeled following Integrated Definition Method #0 (IDEF0) (National Institute of Standards and Technology 1993).

IDEF modeling provides representations for both processes and functions (National Institute of Standards and Technology 1993; Mayer et al. 1995). The Integrated Definition Method #0 or IDEF0, illustrated in Figure 3.5, provides a framework for developing functional models that can be used to define how elements such as people, information, software, raw materials, etc. work together to perform an operation (National Institute of Standards and Technology 1993). Functional models generated within the IDEF0 framework are expandable. Following the Integrated Definition Method #3 (IDEF3) (Mayer et al. 1995) process descriptions may be added to describe sequencing of events. These models, like those created following UML or SysML, are information-centric and provide a systematic approach to modeling highly complex systems.

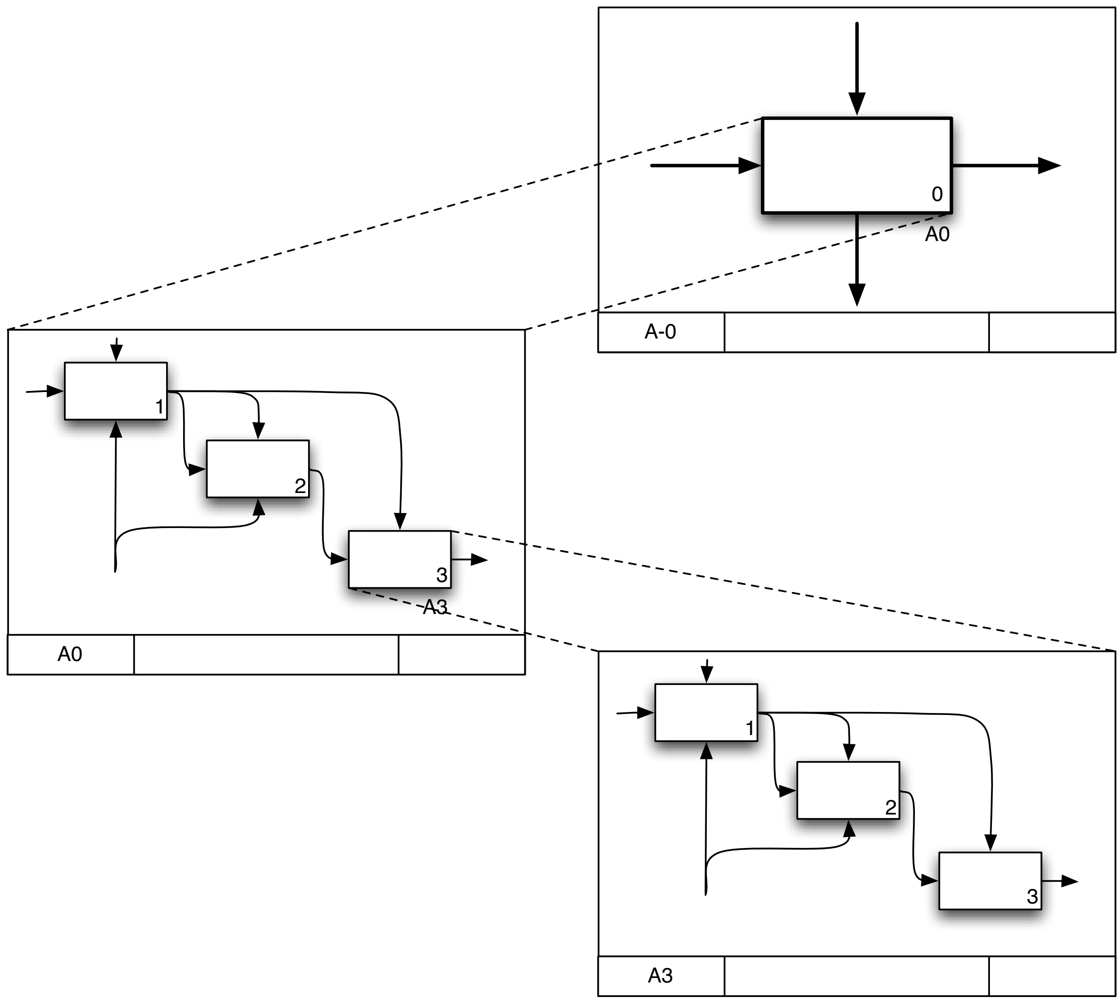


Figure 3.5 Example of the hierarchical IDEF modeling structure (Adapted from (National Institute of Standards and Technology 1993)).

Similarly, modeling languages such as UML (The Object Management Group 2009) and SysML (Friedenthal et al. 2008), which have roots in both computer and systems engineering domains, provide the graphical language and framework to represent requirement diagrams to define system components, behavior, functionality, constraints and requirements. SysML builds, extends and streamlines on the earlier UML (SYSML.org 2008) maintaining, modifying or removing many UML diagrams. The final SysML model structure, represented in Figure 3.6, contains nine diagram types that include: requirement, activity, sequence, state machine, use case, block definition, internal block, parametric, and package (Friedenthal et al. 2008). Of these diagrams, the use case models are most similar to the process modeling used in this research. Use case captures interactions between actors (i.e., the customer) and the system to accomplish a set of goals—this definition is also similar to *jobs* as proposed by the Outcome-driven Method (Ulwick 2005). Use case models are very similar in detail to activity diagrams (Otto and Wood 2001) in that they describe how the actor will interact with the system being defined. These models are information based describing how information must flow between the system and the actor during interactions (Ambler 2009). In activity diagrams, flows represent the movement of the system (i.e., the product) between events that often represent customer-product interactions (Otto and Wood 2001).

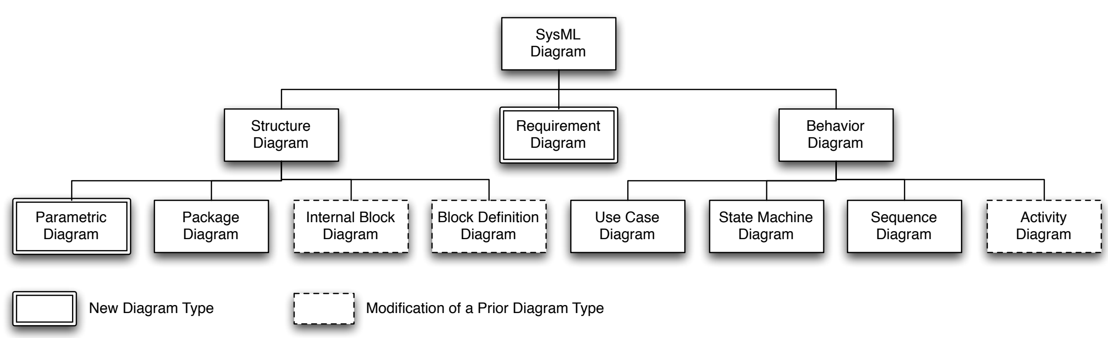


Figure 3.6 Hierarchy of diagrams available in the SysML model structure (Adapted from (Friedenthal et al. 2008)).

The modeling of the user's actions with the proposed product is also comparable to the philosophy of user-centered design that guides human-computer interface design. User-center design broadly covers methodologies where the end-user directly influences the final design embodiment (Chadia et al. 2004). Concern is placed on how the customer will use the final design, and the customer's needs, wants and constraints are considered through all phases of the design process. This idea parallels Ulwick's of considering the functional jobs of the customer through the design process.

In engineering design, user-driven designs are often considered those where the functionality or aesthetics of the interface drive the design process. These designs often have a higher percentage of user interaction than technology driven products (Ulrich and Eppinger 2004). Norman, in *The Design of Everyday Things*, gives four guidelines to center the design process around the user: (1) Make it easy for the user to understand all possible actions at any moment while using the product; (2) Make all things visible; (3) Make the product's state easy to evaluate; (4) Map natural actions to required product actions (Norman 2002). Cagan and Vogel advocate an analogous User-Driven Process that places the user experience at the center of the product design process (Cagan and Vogel 2002). This user experience is influenced by expectations of product value, aesthetics and performance. The use of scenarios facilitates the understanding of the customer's expectations. Similarly, Affordance-based Design (Maier and Fadel 2003) focuses on the various uses offered by a product to user groups. These uses are placed at the center of the design process, and identified affordances are then used to drive concept generation and embodiment of the design. Affordance-based Design builds on the idea of affordance, first coined in psychology by Gibson (Gibson 1986) as a way to describe what is offered to an animal by its environment, and then explored in product design by Norman (Norman 2002) as a way to describe what a product offers to the customer.

These approaches, however, lack mechanisms required to link with function-based abstractions for use during conceptual design activities. The scope of this research is in the first two phases of the design process: Understanding Customers and Conceptual Design. Therefore, a process modeling methodology based on functional modeling with the Functional Basis (Hirtz et al. 2002) is chosen to represent customer-product interactions (Nagel, Hutcheson et al. 2009). This approach to modeling processes pairs with functional modeling and may be applied during the conceptual design phases. The result is a modeling structure that captures product-centric events (similar but more structured than activity diagrams (Otto and Wood 2001)), interactions with the product (similar to use case models (Ambler 2009) but with the benefit of capturing material and energy interactions), and changes to the overall structure of product (Nagel, Hutcheson et al. 2009).

Process models generated during design are useful in that they allow a designer to consider how a customer will interact with and use the product. There are countless possible interactions; a few might include: Changes to the structure of the product, interactions for use of the product, maintenance related decomposition of the product, or configurations for the storage of the product. For example, an event—clean kitchen gadget—may require configurations describing the disassembly of the product into dishwasher safe and hand wash only components. In this way, a process model extends functional representations to capture customer needs related to expected use of a product.

3.2.2 Process Modeling Methodology

Process modeling based on the Functional Basis provides a technique to abstract the operations performed by customer on product during the conceptual design phase. These abstractions provide a means to detail changes to product configurations through a series of events occurring over time. Configurations directed toward a common end goal create an event, and events combine to a black box to create the hierarchical relationship. Overall the collection of the black box, events and configurations abstract the operations expected of the final customer such that he or she may achieve the desired outcome defined by a customer needs set. The model layers that comprise a process model include:

- **Black Box (Process) Model:** The high-level process model defined by a single overall event representing the task to be accomplished.
- **Event Model:** A more detailed process model consisting of multiple events that collectively define the customer's operations with the product.
- **Configuration Model:** A detailed model of the individual actions and changes occurring to the product as a whole and involved in completing a particular event.

The decomposition of an overall process into a process model follows a methodology similar to that of the generation of a functional model. Generally, the following six steps are followed:

1. Identify the overall process associated with the product being designed. Identify the requirements—including the customer needs,

tasks, goals and outcomes—for the process and product as well as the available material, energy and signal flows.

2. Formulate a black box model for the process being modeled based on the requirements identified during Step 1. This model defines the overall process for the usage of the product being designed along with its associated energy, material and signal inputs and outputs.
3. Identify or formulate the events necessary to complete the current or proposed process using the requirements identified in Step 1. For each event, identify input and output flows, start and stop times, and required product configurations such that the product can be transformed to meet the customer's specific requirements.
4. Formulate the event model to consist of chains of events that must be completed systematically to achieve the desired goal or outcome. The event sequence should begin with the initial product action and should be followed by all other discrete events including each of the actions, environments or situations where the product will be used over time. The product should be included as a flow through each event.
5. Decompose each individual event in the process model into a configuration model detailing the discrete changes to the product and any associated functional interactions with other flows in the event. As necessary, time should be represented as either a time flow or as a time line. If a time line is used to represent the flow of time, signal flows may be used to connect each configuration to the time line.

6. Verify that the process models generated address all of the customer's requirements identified in Step 1, abstract all expected operations and model the achievement of the customer's expected outcomes stemming from the application of the product being designed.

Like a functional model, the highest level of detail is the black box model. The black box process model defines the overall process that the customer expects to accomplish with the product being designed. It is derived from the requirements for the process including the customer's overall process requirements, needs and tasks, the process goals and the desired outcomes. The black box process model, shown in Figure 3.7, is created similarly to a black box functional model. It represents a single event with input and output flows to identify all elements required to complete the process, start and stop times and the product. As with functional modeling, materials are bold arrows, energies are single weight arrows and signals (including time) are dashed arrows. These three types of flows are drawn entering and exiting the black box model of the process.

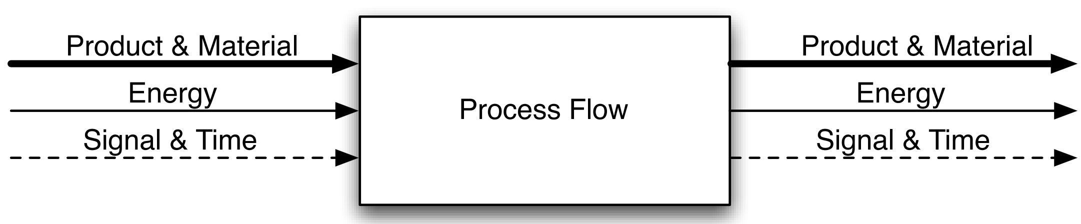


Figure 3.7 Example black box process model.

An important distinction between process and functional models is the inclusion of the product in the process model as a flow. With functional models, the product is not included as a distinct flow; the focus of such models is the set of transformations occurring inside the product and the product cannot act upon itself. With process models, however, the focus of the abstraction is outside of the product, and it is important to include the product as a flow. The inclusion of the product as a flow allows the designer to explicitly model customer and environmental interactions with the product as well as product configurations. This is a primary distinction between traditional functional modeling techniques and the proposed process modeling technique.

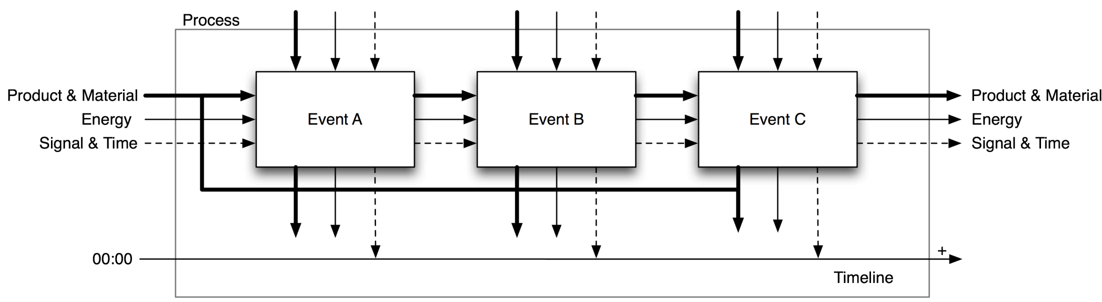


Figure 3.8 Example event model for the black box of the process.

Once the individual events and their required flows are identified, chains are generated. Each element in the chain resembles a single black box containing input and output flows of materials, energies, and signals as demonstrated by Figure 3.8. The first event of the model starts with the initial action or operation of the product, and each progressive element in the event model identifies new operations that must occur as time progresses. Temporal information is captured with either signal flows or a time line. A time line is drawn parallel to the process model with the initiation time at the left and the completion time at the right. Times marked along the time line are tied to each event with a dashed signal flow. Figure 3.8, however, shows time in both formats for illustrative purposes.

In an event model, there are inter-event flows that are required for more than one event and intra-event flows that are required for only one particular event. Inter-event flows are typically drawn entering and exiting the vertical (left and right) sides of an event box, where intra-event flows are typically drawn entering and exiting the horizontal (top and bottom) sides of an event box. Flows can also skip an event or feedback to an earlier event depending on the customer needs and requirements. Figure 3.8 shows a material flow feedback between the third and first events.

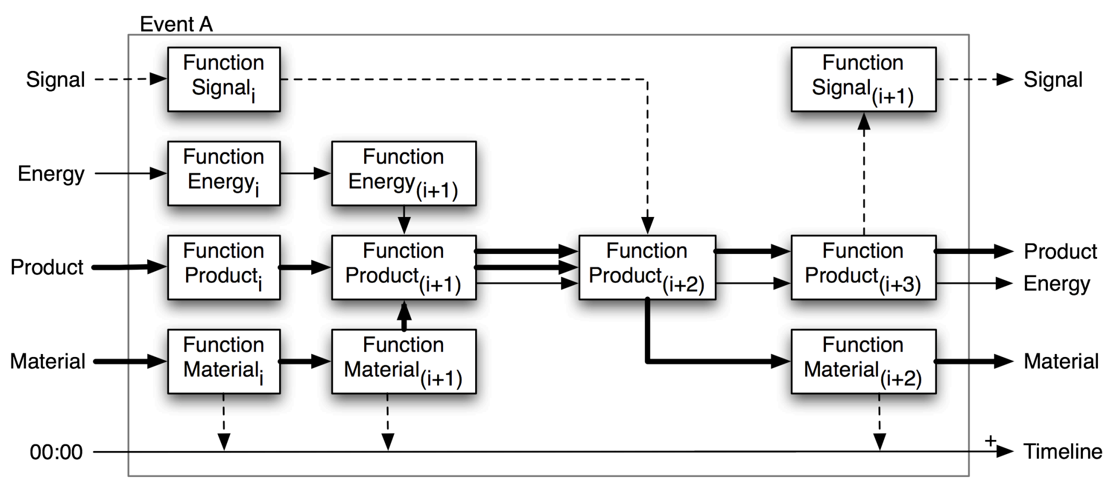


Figure 3.9 Example configuration model for an event.

Configuration models, developed for each event in the process, are the most detailed level of a process model. Configuration chains should be made for all of the input flows (materials, energies and signals) as well as for the product. Each chain should capture all of the individual changes to each flow and product configurations that must be achieved in order to arrive at the desired output(s). Once each of the chains is produced, they are aggregated to create a complete configuration model like shown in Figure 3.9.

3.3 Summary

Process models provide an extension to functional modeling and are intended to be the starting point for functional analysis. Their generation should lead the design team toward further insight on the required functionality of the product being designed. With process models, the boundaries between how a product is used and what a product does begin to blur. Depending on how the product's boundaries have been defined, functions may be shared between sub-functional and configuration models as illustrated in Figure 3.10.

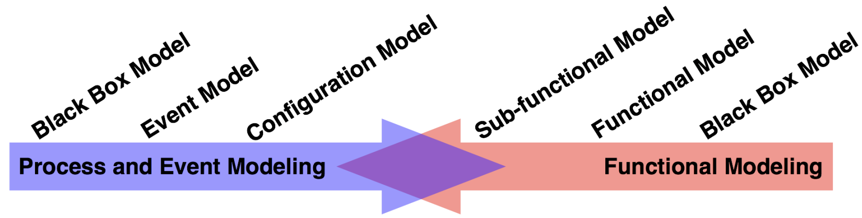


Figure 3.10 The relationship between functional and process models.

This relationship between function and process occurs because of their natural hierarchy. This is most evident with the modeling of flows from the environment that interact with the product yet are expected for successful operation. Environmental flows may be imported into both the configuration and sub-functional models of the system and have functionality in both models dealing with the flow. It is important to remember, however, that the overall perspective between a process and function is different, and this difference should be reflected in the modeling of the flows shared between configuration and sub-functional models.

CHAPTER 4 Integrating Functional and Process Modeling

Flow-based process modeling (Nagel, Hutcheson et al. 2009) and functional modeling with the Functional Basis (Hirtz et al. 2002) may be integrated to provide depth and breadth to models by using a hierarchical approach to model generation. Both functional and process models exist at different levels of fidelity and complexity depending on the problem being addressed. The level of fidelity is defined by the designer generating the functional model and may be divided for breadth via process modeling as illustrated by Figure 4.1. Process models, generated in conjunction with functional models, allow for formal functional model segmentation based on changes in required functionality between various anticipated operations of the product being designed.

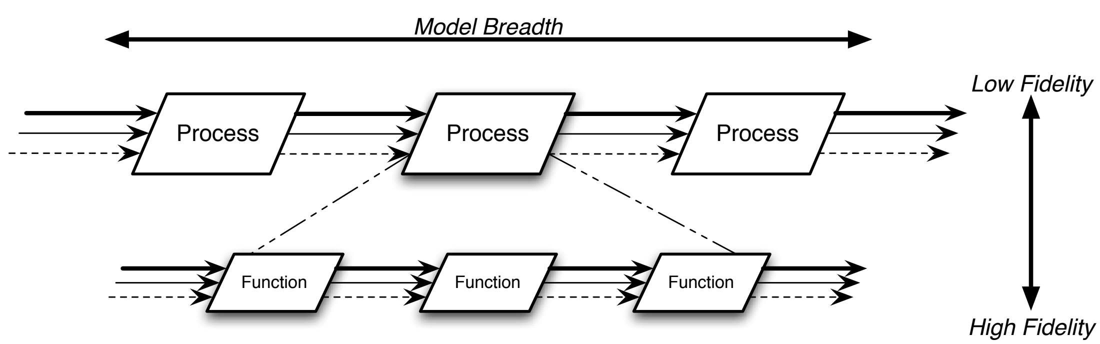


Figure 4.1 Representation of processes providing model breadth and functionality providing model fidelity.

Process models and functional models generated in this way place the product being designed as a part of the process. Customer actions and changes to the product are captured in the process layers of the model where the product is modeled explicitly as a flow. The functional layers of the model increase the fidelity of the process as illustrated in Figure 4.1. Functional layers capture the flow transformations required for product operation and internal to the product. As with functional models, the number of levels required to adequately represent the process model of a system depends on the complexity of the system being modeled and the specific design goals.

4.1 Approach

The approach to integrated model generation consists of three phases that increase in model fidelity and narrow in breadth as illustrated in Figure 4.2. The phases may be applied independently to create a stand-alone model or may be applied in order to create functional and process models integrated by flow requirements. The three phases are: (1) Define the environment, (2) Define relevant processes and (3) Define functionality.

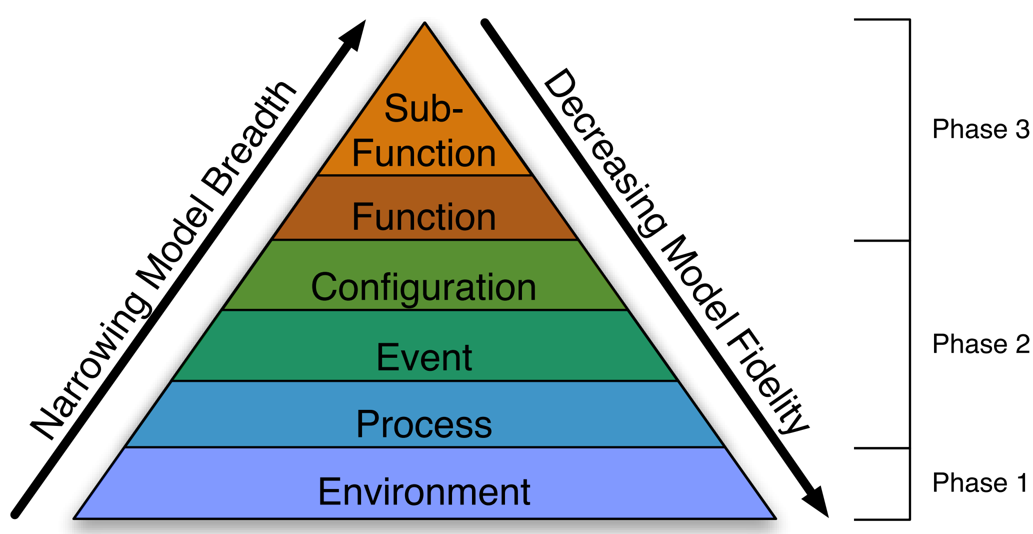


Figure 4.2 Illustration demonstrating how the levels of integrated functional and process models correspond to the problem scope.

Phase 1: The environmental boundaries for the product are defined by first identifying the customer needs for the product being developed. The environmental boundaries capture where the final product will operate and encompass all of the functional and process models providing the external boundary from which all flows required by the product must originate.

Phase 2: Processes define the situations where the product will be used. These models consist of the operations, changes and actions of the product as a whole. To capture this information, the product is modeled as a flow, and other flows in the process can act on and transform the product.

Phase 3: Functional decomposition captures flow transformations within the product; these transformations define what the product must do and are decomposed to an appropriate level to capture the identified customer needs.

Each of these three phases focuses on capturing the entire product usage cycle and developing depth that corresponds to the problem scope. This concept is illustrated by Figure 4.2. During phase one, the environment provides the most broad model representation capturing the least detail yet the broadest scope. The environment comprises all model elements although it has the lowest fidelity. The process decomposition defined during phase two increases in fidelity by capturing more detail on how the product will be used by the customer. The process layer contains a black box, events and configurations as defined in Chapter 3. In the third phase, models focus within the physical boundaries of the product via functionality—also as defined in Chapter 3. First a high-level function, in the form of a black box, defines the overall purpose of the product. Then with the sub-functions that decompose this model, flow transformations represent what must happen within the product.

Sub-functional models have the most fidelity, yet the least breadth. Sub-functions capture the most specific details of how the product should operate.

The exercise of identifying the scope of the design problem with the creation of integrated functional and process models is analogous to answering the five Ws in investigative reporting. Following the five Ws formula, the reporter seeks to answer five questions: *Who* was involved in the event? *What* happened? *Where* the event took place? *When* the event happened? And, *Why* the event occurred? In addition to these five questions, *how* the event occurred is often considered resulting in *five Ws* and *How* (Jones 1976). In Cagan and Vogel, the same five Ws are applied to product design to develop a scenario describing how the lack of a product makes the completion of a task more difficult. To elicit information, the questions asked are: "*Who* is the target customer? *What* is their need? *Why* do they have that need? *How* is the task currently accomplished? [And,] *When* does this happen?" (Cagan and Vogel 2002).

Applying the five Ws to each of the three phases for creating integrated models aims to answer similar questions. First, the customer and their needs are identified defining ***who*** will use the final product. Then as modeling begins, the environment models answer ***where*** the customer will use the final product. Process models answer ***why*** by capturing the reason the customer will require the product. Adding a time line to the process representation answers the question of ***when*** by defining time constraints placed on the use of the product. Then as model representations focus within the boundaries of the product to model function, the question of ***what*** is answered. Function defines ***what*** the product must do to operate within the expected configurations, events and environments defined from the needs of the customers. Following these three modeling and abstraction phases, the design team moves on to answer the final question—***how***. Answering the question of ***how*** moves the design team into the next step of the design process to consider components and solution strategies that answer the previous five Ws.

4.2 Methodology

The decomposition of these phases into specific steps leads to the six step methodology shown diagrammatically in Figure 4.3. The methodology begins just after the customer needs for a product have been identified. In step one, the customer needs and goals are analyzed to create a representation of the environment where the product will be used. A process model (within the environment) is next derived to define how the customer will use and interact with the product. Definition of the process begins with the formulation of a black box process model. The black box process model is decomposed into a chain of events (represented within the high-level black box process), and each event is further decomposed into configurations (within each event). The configurations define high-level changes to the product as a whole. The collection of configurations is equivalent to a black box functional model for the product. Black box and functional models may be independently generated for each configuration change modeled during the fourth step. Typically, however, a single functional model is created for the product. This functional model aggregates all operational aspects of a product design into a single black box functional model and its decomposition as represented in Steps 5 and 6.

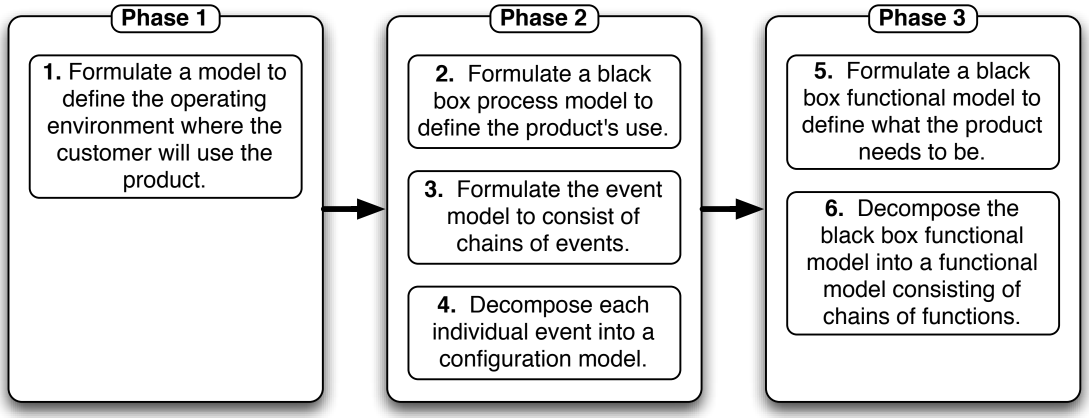


Figure 4.3 Methodology for integrated modeling.

The final abstraction can be mix of environment, process and function decompositions interconnected in such as to best represent the system being designed. For example, consider an automobile, a cold and snowy environment might require different processes and functionality from a warm and humid environment. These different processes and functions may be separated by the environmental conditions they are in place to fulfill. Also, each of the three phases may be applied independently to create a stand-alone model of the environment, a process model of the product usage or a functional model of what the product must do.

4.2.1 Defining the Environment (Phase One)

At the top level of the modeling hierarchy, models capture the environment where the product will be operating. During the first phase, the modeling scope is the most broad, with modeling elements defining ***where*** the product will be used. Customer needs scope the design problem. The first step in the methodology relates to the formulation of an environment model:

Step 1: Formulate an environment model from the customer needs to define where the customer will use the product. This model defines the location where all flows will be derived as well as all operating conditions to which the product will be subjected.

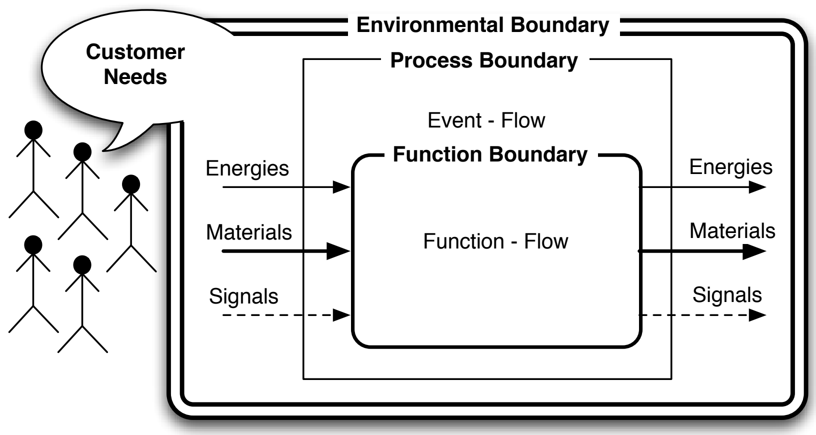


Figure 4.4 Representation of environment, process and function.

The environmental models are in essence boundaries or bounding boxes. They are comprised of the product and all flows required for the product to be operational. The operational environment must be set to surround these defined flows. Flows that cannot be obtained directly from the environment, but are required for the operation of the product, will need to be generated through the processes or within the product itself. In the context of a process model, the product will exist as an explicit flow as well. This concept can best be represented in the form of a graphic as shown in Figure 4.4 with the outermost, double-lined, bold box. In Figure 4.4, the process occurs within the environment, and function is a part of the process. Customer needs define all three model layers, and should directly influence the model elements and structure. The flows of energy, material and signal that come from the environment are used during all model levels.

4.2.2 Defining the Process (Phase Two)

Once the environment has been established, the second phase, defining the process, is begun. During the generation of process models, the breadth of the design process is narrowed with the events and configurations answering ***why*** the customer will purchase the product and ***when*** the operations will occur with either an explicit timeline or the ordered operation of events.

Events are modeled as two levels: First, with the black box process model capturing the high-level task that the customer will use the product to complete. Second, with the event model capturing the breakdown of individual operations required to complete the high-level task. The event model takes the form of a chain of individual events connected with input and output flows. Flows follow the energy, material and signal convention and include the product being designed as a flow. Steps two and three guide the development of event models:

Step 2: Formulate a black box process model from the overall customer goal identified during the collection of customer needs. This model defines the overall usage of the product being designed along with its associated energy, material and signal inputs and outputs.

Step 3: Formulate an event model to consist of chains of events connected by flows of materials, energies, and signals that must be completed systematically to achieve the desired goal. The event sequence should begin with the initial product action and should be followed by all other discrete events including each of the actions, environments or situations where the product will be used over time. The product should be included as a flow through each event.

Once events are generated, each individual event where the product will be used is decomposed into a configuration model. The configuration models detail the discrete changes to the product and any associated functional interactions with other flows in the event. The decomposition begins by first generating chains of configurations for each input flow identified for the event being modeled; inputs are transformed into outputs to detail the flow-to-product interactions required. Configuration chains are aggregated together to create a complete configuration model. Step four directs the generation of the configuration models:

Step 4: Decompose each individual event in the process model into a configuration model detailing the discrete changes to the product and any associated functional interactions with other flows in the event. Begin by generating chains of configurations. Once configuration chains capture changes from input to output, aggregate chains into complete configuration models.

To more fully illustrate this decomposition consider the operation of an automobile. The driver may have a high level process to operate a car, but during that process of operating the car, different events may occur—some normal such as cleaning, fueling, driving or loading cargo and passengers. Some, however, might be abnormal such as a crash. Each of these events are also associated with configurations or changes that must occur to the car as a whole. Without these configurations, the event could not occur. An example might be the compression of the crumple zone during a crash or opening the fuel door to access the filler neck for fueling. During this second phase, these changes are left as high level interactions with the product that achieve the desired event. Configuration changes only describe changes to the whole product.

4.2.3 Defining the Functionality (Phase Three)

Functional models explore the transformations required within the product. They provide a high-fidelity description of ***what*** the product must be. Functional models, like with the relationship between event and configuration models, tend to be modeled at two levels: First, with the black box functional model capturing a high-level description of what the product is supposed to do, and second, with a functional model capturing the internal flow transformations occurring on each input to create the desired outputs. There is, however, no rigid statement that functions must contain no more or no less than two levels of detail.

Like the environment and process models, the functional model follows the energy, material and signal flow convention. Since, however, the product cannot be a part of itself, it and its components should not be explicitly included as flows within the functional model. To generate the functional decomposition, Steps 5 and 6 are followed:

Step 5: Formulate a black box functional model to fulfill the circumstances abstracted by the process model. Functional models may be independently generated for each configuration block modeled in Step 4. Typically, however, a functional model is created for a single configuration block that represents all of the operation aspects of the final design or an aggregation of all operational aspects of the final design. The decomposition path is a choice that is left to the design team. Flows required in the functional model should be derived from flows in the process model.

Step 6: Decompose the black box functional model into functional models detailing the discrete changes to each flow within the boundaries of the product. Begin by generating chains of functions. Once function chains capture changes from input to output, aggregate chains into a complete functional model.

Black box functional models abstract the high-level changes that the product must be designed to perform. Black box and sub-functional models may be independently generated for each configuration change; typically, however, a single functional model is created for the product focusing on a few key configurations or a key event. Most frequently, this is the *operational* event. The sum of the flows entering and exiting the product through the configuration models represent the minimum flows required by the product.

The black box functional model is decomposed following Step 6. The decomposition begins by detailing the discrete changes to each flow within the boundaries of the product. First chains of functions are generated for each flow capturing changes from input to output followed by the aggregation of function chains into a complete functional model.

4.3 Example: A Bridge Kit Student Design Project

To illustrate the creation of integrated functional and process models, consider a bridge kit student design project that was assigned during a design class at the Missouri University of Science and Technology as a case study.

The customer needs for the bridge kit stated that the bridge must be constructed quickly by a single person from an initial configuration no larger than a cube of 0.25 m^3 for the crossing of a 2.15 m ditch in Schuman Park, a local city park. The kit, once constructed, must be positioned and secured over the ditch, and once positioned, a person must cross the bridge without falling into the ditch. These high level needs may be summarized as: (1) *The bridge must be constructed at a remote location from a self-contained kit.* (2) *The bridge must be transported from the construction site to a ditch where it will be positioned for crossing.* (3) *The bridge must support a team member to allow for ditch crossing.*

4.3.1 Defining the Environment (Phase One)

From the customer needs for the bridge kit, the environment is selected as *Schuman Park Ditch*; the flows required include the *bridge kit*, the user—modeled as *human material*—who will construct and cross the bridge, the user's *human energy*—required for construction, positioning and locomotion—and the *ground* including the ditch where the bridge will be positioned. These flows have been drawn in an example environmental boundary for the bridge kit in Figure 4.5.

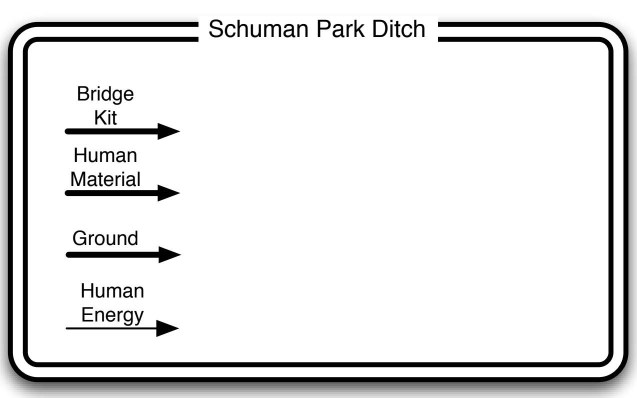


Figure 4.5 Example environmental boundary for the bridge kit project case study.

4.3.2 Defining the Process (Phase Two)

For the bridge kit example, shown in Figure 4.6, three high-level events are identified. Since the bridge initially is configured as a cube of 0.25 m^3, its assembly is required to create a bridge spanning 2.15 m; this process is modeled as the event, *construct bridge*. The second step required of the user, is the positioning of the bridge; this is simply named *position bridge*. Finally, once the bridge is in place, the problem statement given to the students states that the bridge must be crossed; this has been named *cross bridge*.

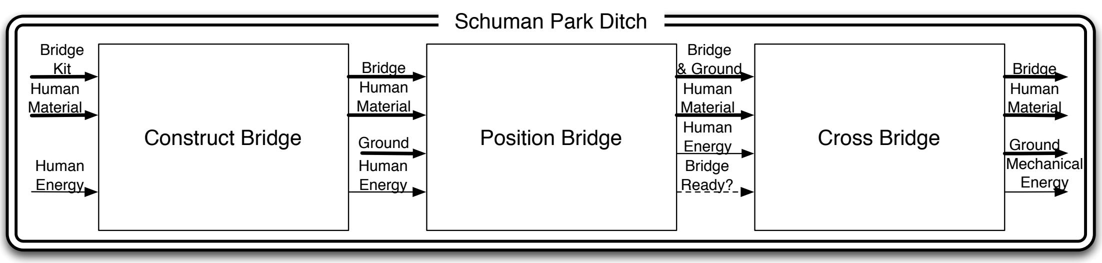


Figure 4.6 Example black box process models for the bridge kit project case study.

Flows of the bridge kit, human material and human energy enter the construct bridge event as defined by the customer need, "Bridge must be constructed by the team." Once the kit is converted to the bridge, it travels into the position bridge event along with the ground, human material and human energy. The bridge once positioned joins the ground flow for the final event to allow the users to cross the bridge. A ready signal is generated to indicate that the bridge is ready to cross.

Three potential event model decompositions are generated—one for each event. These configuration models, generated for the *construct bridge*, *position bridge* and *cross bridge* events are Figures 4.7, 4.8 and 4.9, respectively.

The *construct bridge* configuration model in Figure 4.7 describes the bridge kit as materials being brought into the system. The users separate the materials and convert them into a fully assembled bridge. The bridge and users then leave the *construct bridge* configuration model.

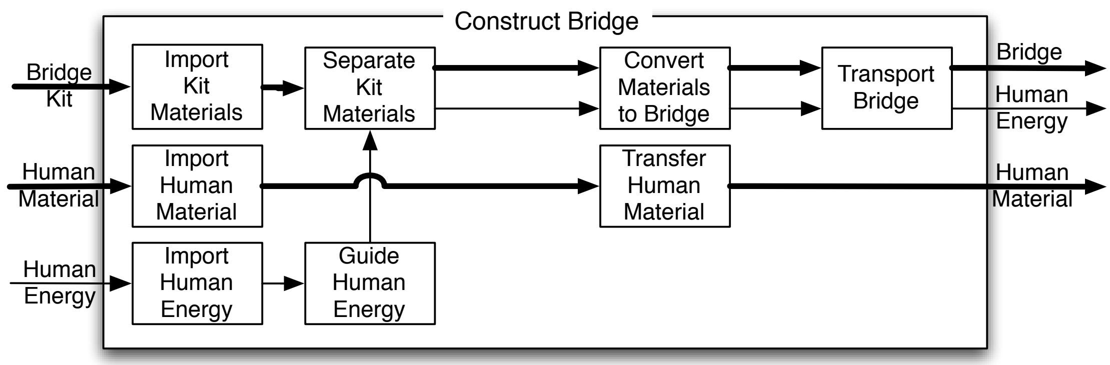


Figure 4.7 Configuration model decomposing the construct bridge event for the bridge kit project case study.

This bridge is next positioned over Schuman Park ditch. The configuration model in Figure 4.8 describes what the user will do with the bridge during the *position bridge* event. The users guide the bridge into place and position it for coupling and securing. Once secured, a *ready* signal is generated to signal the users that the bridge is now safe to be crossed.

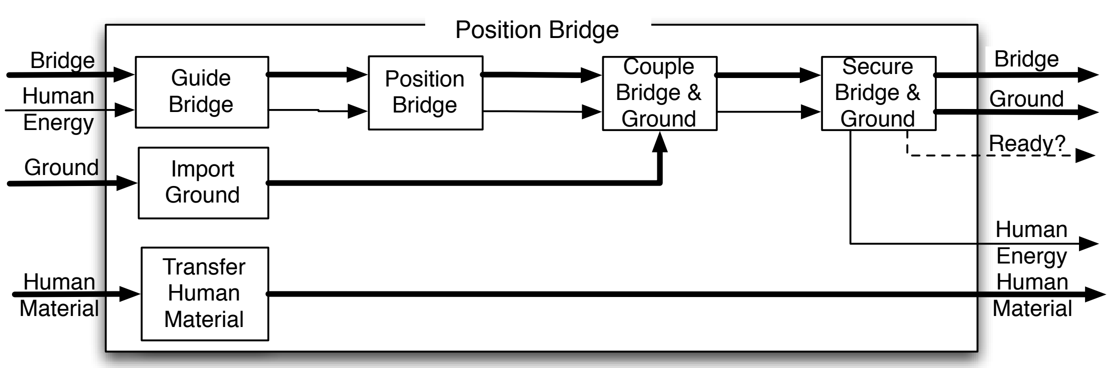


Figure 4.8 Configuration model decomposing the position bridge event for the bridge kit project case study.

The configuration model for the event, *cross bridge,* shown in Figure 4.9, describes the human being guided to the bridge and the bridge providing the support required for the human to cross the ditch. Once the bridge is crossed, the human and bridge separate and leave the process.

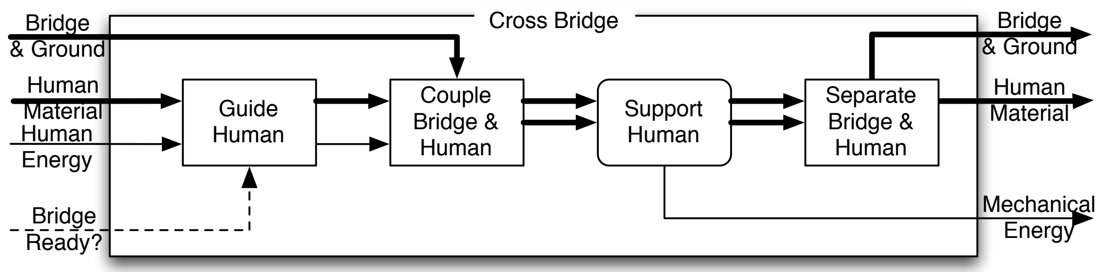


Figure 4.9 Configuration model decomposing the cross bridge event for the bridge kit project case study.

4.3.3 Defining the Functionality (Phase Three)

For the bridge example, the functional decomposition chosen for the bridge focuses on the *cross bridge* event. Within this event, the black box functional model for the bridge, *support human,* is housed as illustrated in Figure 4.10. The flows entering the event are extended to enter and exit the black box functional model as well. Further functional boundaries could be defined for the two remaining process boundaries, but the design team must decide how best to capture the customer needs of the product. When modeling the bridge, the design team decided that the remaining two events are fully captured by the configuration models since they deal primarily with the assembling and positioning of the bridge as a whole instead of internal changes to the bridge.

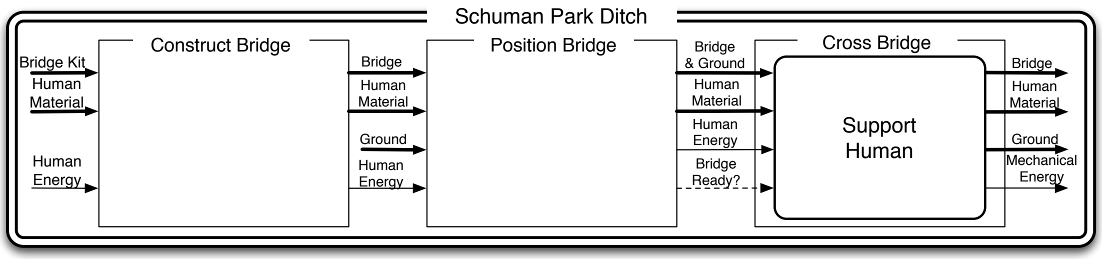


Figure 4.10 Example functional black box (Support Human) for the bridge kit project case study.

The *support human* black box decomposed functionally in Figure 4.11 captures the human material being guided across the bridge when the ready signal is present. Human material is converted to mechanical energy as they are supported by the bridge; this mechanical energy is guided into the ground as the bridge supports the human's weight. Human material is then exported from the system once the ditch is successfully crossed.

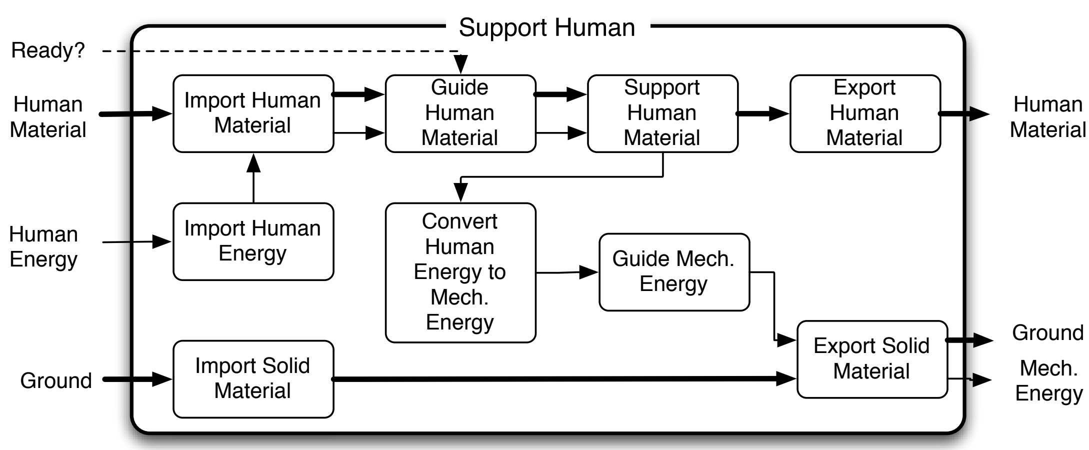


Figure 4.11 Example functional decomposition of Support Human for the bridge kit project case study.

4.4 Focusing Model Decomposition

Model decomposition may initially focus on either the product's functionality or the process in which the product will be used depending on the requirements of the design task. A process focused decomposition centers on the generation of configuration models to detail the specifics of the operations occurring on the product. A function focused decomposition centers on the flow transformations that occur within the physical boundaries of the product.

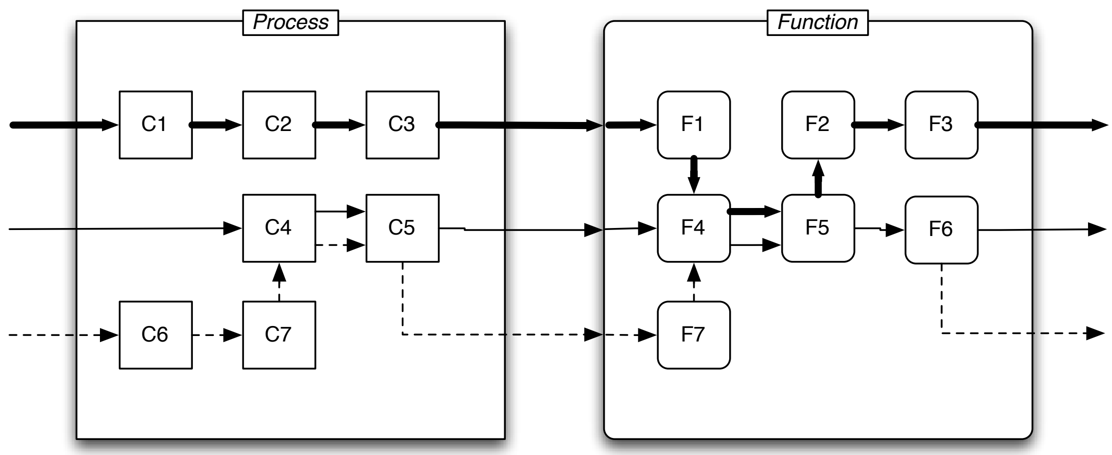


Figure 4.12 Example aggregated process and functional model where C-blocks represent configurations and F-blocks represent functions.

Decompositions, whether of functions or processes, are based on the transformations of material, energy or signal flows. These transformations may first be generated independently as chains or *chunks*, but to capture how flows interact, they must be aggregated. This aggregation represents how different elements work together to meet the expected outcomes and is the key to creating integrated process and functional models. Functional models and process models are aggregated together to create a common description of the product being designed as illustrated by Figure 4.12. When customer needs can best describe ***why*** the customer will use the product, then process-based decompositions should be used, and conversely, when customer needs focus on ***what*** the product must do, then function-based decompositions should be used. The division between these aggregations is the boundary for the event or function where they are contained. For example, a process-based decomposition may be required to describe how the customer is going to disassemble a product for maintenance and a functional decomposition may be required to describe how the product is going to operate. These two decompositions may share similar flows and occur sequentially, thus they should be aggregated together into a single representation like the illustration in Figure 4.12.

With the bridge kit example, model decompositions have been created for each event and for the functional black box. When the three high-level customer needs (discussed in Section 4.1) are considered, however, it may be deduced that not all decompositions are necessary to fully represent the complete needs set. Two needs focus on the way that the team will use the bridge while the other focuses on the operation expected of the bridge. Aggregating from these needs creates a combined model of function and process, where the event decompositions for *construct bridge* and *position bridge* are integrated with the functional decomposition of *cross bridge.* This combined model, provided as Figure 4.13, aggregates the configuration models with the functional models following the flows identified at the event and/or black box functional level. The bridge kit is first assembled into the bridge and then positioned. The bridge flow enters the *cross bridge* event, but then is decomposed functionally to represent what it must do to support the human as they cross the ditch.

When making integrated models similar to the bridge kit case study, the designer should first consider the environment and process to ensure that the product's scope is not too limited (not meeting customer demands) or too encompassing (becoming cumbersome for a customer and ultimately not useful). Once the process and environment are defined, then either a functional model or more detailed process decompositions are developed based on the appropriate energy, material and signal flows and the derived output energy, material and signal flows. All input and output flows required to develop the process and functional models come from the environment where the product will be used.

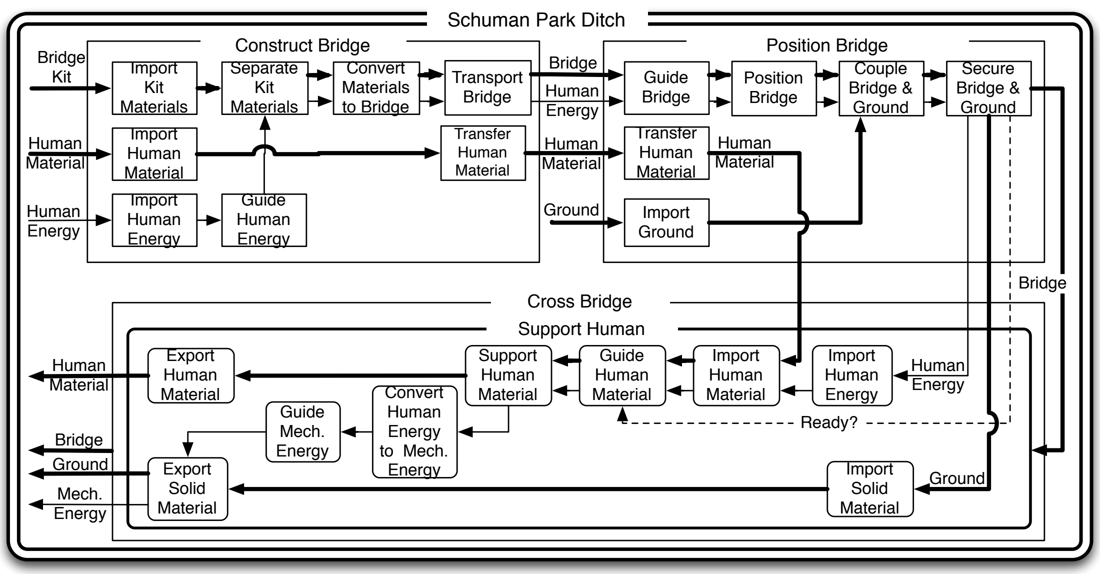


Figure 4.13 Example decomposition of an integrated function and process models for the bridge kit project case study.

For the bridge case study, a traditional functional decomposition would focus inside of the functional boundary and capture the operation of the bridge; this decomposition would be the same as the functional decomposition shown in Figure 4.11 within the *support human* black box. This functional decomposition, however, fails to capture all of the customer needs for the bridge project; the students are required to have the bridge constructed from a kit and be positioned as a whole unit over the ditch. Modeling these elements from the customer needs would require that the bridge be a part of the functional model and would violate one of the primary standards of functional modeling; a product cannot be a part of itself, thus the product cannot be a part of its functional model.

To completely capture the customer needs related to assembly and positioning, process models are used to decompose the *construct bridge* and *position bridge* events. The *cross bridge* process, however, is not used in the aggregated model shown in Figure 4.13. This process model represents the human being guided to the bridge and the bridge providing for the transfer of the human across the ditch. Once crossed, the human and the bridge separate. Since, however the process decomposition is based on the interactions between the customer and the product, it fails to meaningfully capture the needs related to supporting the student as they cross the ditch; these needs relate to the internal structure of the bridge itself. Thus, following standard functional and process modeling, two independent models would be required to capture the complete set of customer needs.

4.5 Computational Tool

To facilitate the creation of integrated models, the computational tool FunctionCAD is developed (Nagel, Perry et al. 2009). FunctionCAD, available for download on the Design Engineering Lab website[i], is a modular, open source application designed to create integrated, hierarchical models consisting of environments, processes and functions. FunctionCAD is written with

---

[i] http://DesignEngineeringLab.org/FunctionCAD/

C++ and uses Qt4 by Nokia (previously Trolltech) for a cross-platform Graphical User Interface (GUI). Files created from FunctionCAD are based on the open outline format OPML (Outline Processor Markup Language) to provide compatibility with existing conceptual design tools. A plugin interface provides an extensible modeling environment with the flexibility to interface with future unanticipated design tools. FunctionCAD is officially supported on MacOSX and Windows, and the source code is available and known to work on Linux.

The basic interface of FunctionCAD contains a tool pallet of six elements for model construction: (1) environment block, (2) process block and (3) function block—corresponding to the three model types defined in FDF—and three flow types of (4) material (represented with a bold arrow), (5) energy (single width arrow) and (6) signal (dashed arrow). The screenshot provided as Figure 4.14 shows an example hierarchal model and the FunctionCAD interface. Model generation within FunctionCAD, is drag and drop. The user selects the block type desired (environment, process or function) and drags the block into the work space. Any block type can be the child or, conversely, the parent of another block type; the only exception to this rule is that the highest level block is constrained to be an environment. To visually represent this model hierarchy, blocks are simply placed within their desired parent. Since this visual representation, can however, become cluttered when a number of children are present, the option to visually "close" and "open" blocks is provided. When a block is "open," its name is displayed top-center and its children are visible; however, when a block is "closed," its name is displayed at block center, and its functional decomposition is hidden. Flows of materials, energies and signals connect the environments, processes and functions. To make a connection, the tip and tail are dragged over the blocks to connect. In Figure 4.14, a single environment contains two individual processes. Each process contains functionality. A second function block contains a two chain sub-functional model. Flows of energy, material and signal connect the blocks.

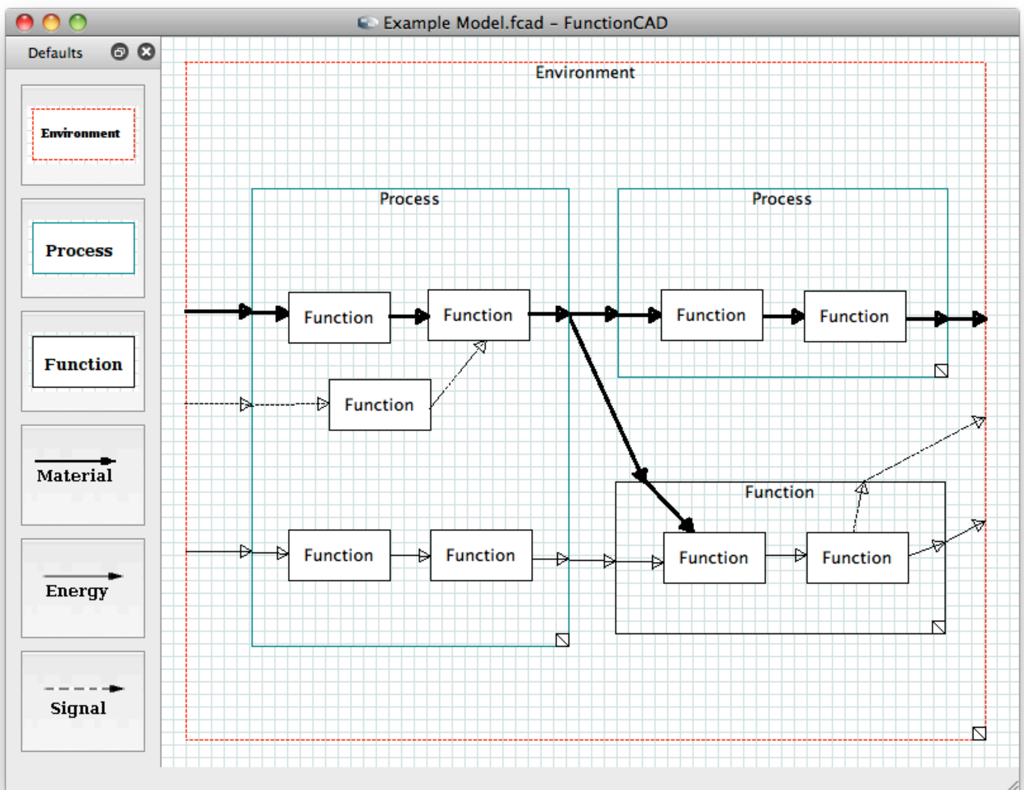


Figure 4.14 FunctionCAD interface with an example integrated model.

4.6 Summary

Integrated functional and process models are generated through three phases to represent the environments where the product will be used, the processes capturing the jobs, tasks that the customer will use the product to complete and the functionality that the product must contain. Modeling through all three phases creates models that have both breadth and depth. Model creation is both flexible and extensible where phases may be applied in order or independently. Integrated models are linked with flows of material, energy and signal to provide traceability between how the product will be used, where it will be used and what the product does. With the 5Ws formula, models are created by answering ***who*** will use the product, ***why*** the customer requires the product, ***when*** the product will be used and ***what*** the product must do. A computational tool may be employed to assist with the creation of these integrated models.

CHAPTER 5 Initiating Functional Design with Outcome-driven Inputs

The generation of functional models, process models or their integration relies heavily on the identification of a proper set of customer needs. Having a complete set of representative customer needs is key to the design of successful products. An appropriate set of customer needs provides the building blocks from which a successful engineered design may be constructed.

Customer needs often drive the engineering design process. For example, in Quality Function Deployment (QFD), the customer attributes are correlated to engineering characteristics through the House of Quality (HoQ) as illustrated in Figure 5.1. Correlations may continue through the design process linking engineering characteristics to parts characteristics, parts characteristics to key process operations and key process operations to production requirements (Hauser and Clausing 1988).

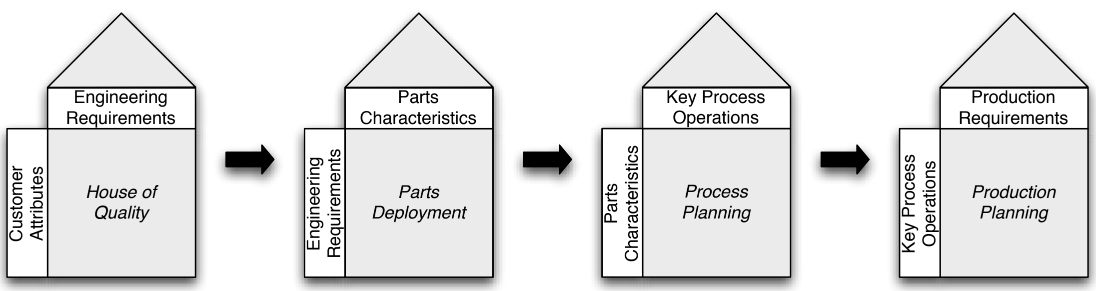


Figure 5.1 House mapping in the Quality Function Deployment method (Adapted from (Hauser and Clausing 1988).

In Axiomatic Design this mapping is similar (see Figure 2.5) with the customer needs—termed Customer Attributes (CAs)—paired to Functional Requirements (FRs), and this pairing continues with FRs paired to Design Parameters (DPs) and DPs paired to Process Variables (PVs) (Suh 1990; Suh

2001). In function and flow-based functional modeling, customer needs are paired to product input and output flows. The transformation of these flows—termed functionality—is paired to solution principles which leads to a detailed and final design implementation (Otto and Wood 2001). Each of these design methods use the customer needs as inputs. These inputs are evolved through requirements and design parameters into a final implementation.

Using customer needs as "customer inputs" to the engineering design process, however, has its limitations. A number of techniques such as focus groups, interviews, questionnaires, being the customer, and ethnographic or observational studies (Otto and Wood 2001; Ulrich and Eppinger 2004; Ullman 2010) may be used, but knowing which inputs to collect and which customers to collect them from is still an issue. Collection of customer inputs often focuses on trying to collect the Voice of the Customer—a hierarchical set of qualitative customer inputs in the customer's own words and their quantitative importance to the customer (Griffin and Hauser 1993). The validity of any quantitative relationships drawn from customer needs has, however, been called into question (Olewnik and Lewis 2007), and others have not found strong correlation in the level of customer satisfaction between collecting the Voice of the Customer versus free-style customer interviews (Stank et al. 1997). The Voice of the Customer is, however, still very important. The key Leonard argues, is "discerning the difference between what customers are able to say and what they want" (Leonard 2002).

To hone in on these important customer inputs, Katz explains that first the scope (i.e., what the design will and will not be) should be identified (Katz 2001). Christensen and Reynor agree, proposing that the critical unit of analysis should not be the customer, but, instead, the circumstances should be analyzed (Christensen and Raynor 2003). Circumstances represent design opportunities where customers "hire something or someone to do the job as effectively, conveniently, and inexpensively as possible" (Christensen and Raynor 2003). Ulwick similarly argues that customers should be asked what delivered outcomes they desire from a new product (Ulwick 2002). In his text, *What Customer's Want*, Ulwick points out that not only do companies not know which inputs they should be obtaining from customers, the customers do not know what inputs they should provide the companies. Thus, the question becomes, why do we ask customers what they want? Ulwick's proposition is instead to use the *jobs* customers wish to complete, the *outcomes* customers use to rate job performance and the *constraints* that limit customers from purchasing products (Ulwick 2005) as inputs to the design process. These inputs are meant to answer why a customer will buy a new product.

In this chapter, Ulwick's Outcome-driven Method (Ulwick 2002; Ulwick 2005) is explored as a way to drive function-based design. The inputs are explored as a starting point to define a process model. Key goals include: (1) Providing rigorous mappings between customer inputs and functionality for function-based design. (2) Improving the completeness of customer inputs represented in function-based design. (3) Facilitating the leap from customer inputs to functionality.

5.1 Approach

Mapping the Outcome-driven Method to integrated functional and process modeling formalizes the leap from customer inputs to abstractions for concept generation. Following the Outcome-driven Method, the three key inputs—jobs, outcomes and constraints—are elicited during customer interviews. These inputs are then used as a starting point to drive the creation of process models. As illustrated in Figure 5.2, jobs are first mapped to the black box process. Ancillary tasks (*sub*-jobs) are mapped to events. Since the outcomes are the metrics a customer uses to judge the performance in completing a job, they map to not only the high-level job, but also to the ancillary tasks (events) and to the flows connecting each of the events. Constraints limit the flows that can be used to connect each of the ancillary tasks, and consequently, they indirectly map to the flows available for the black box process. Therefore, considering both the outcomes and constraints will provide the set of flows required and available to the process.

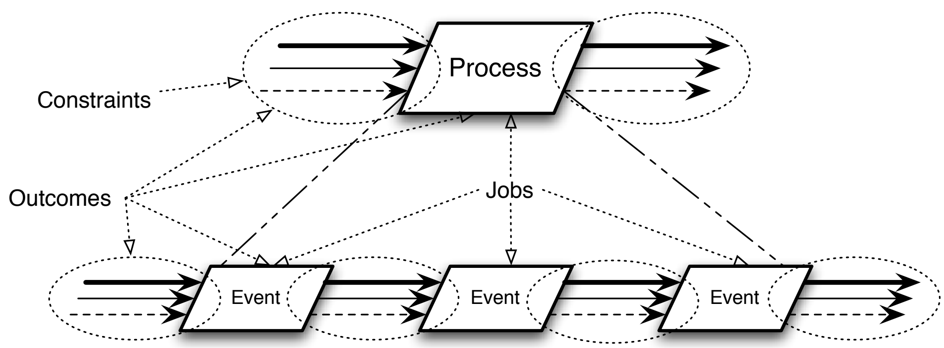


Figure 5.2 Job, outcome and constraint mapping to processes and flows

First, the environments where the product will be used are considered. Environmental boundaries are created to define where the customer will use the product. The customers' expected use of the product is abstracted using the black box process, events and flows. The black box process is placed within this environmental boundary, and all flows required for the process are drawn entering and exiting the black box. Event models are created as a chain of sequential events connected by the flows required for each task. Since process models abstract customer-product interactions, the product being designed is included as a flow. Each event is decomposed into a configuration model to capture the functional changes to the product as a whole.

Models generated in the process layer are meant to capture all activities of a product—first at a high-level with the events or jobs describing *where*, *when* or *why* the product will be used (and by *whom*) and second with detail at the configuration level describing the specific functional instances of the product as a whole. To move from the process layer to the functional layer, the flows required for the product must be identified. The black box functionality must be addressed during process modeling. The flows required during the functional operation of the product should be a subset of those identified during the process modeling step to represent the operation of the product. These flows are extracted from the process layer, and, in conjunction with the black box functionality, are used as a starting point to generate the functional model.

5.2 Methodology

From the research approach, the following methodology is developed to guide a design team to concept generation activities for both redesign and original design cases. The methodology adapts the opportunity identification steps from the Outcome-driven Method to begin. These inputs are translated to process modeling and used in process model generation. Function is extracted from the process models to create a functional model for concept generation activities. Specific steps are outlined below:

1. Gather customer needs using a technique such as focus groups, interviews, ethnographic studies, or surveys to create a compiled list of customer needs statements. In particular, the objective is to classify the customer need statements as either specific *jobs*, *outcomes* or *constraints* for the identified product opportunity.
2. Formulate the raw customer need statements into specific statements of jobs, outcomes and constraints. Typically, these customer need statements are representative of the outcomes and constraints associated with the jobs the product is to perform. *Jobs* are formulated as functional (task to be completed) or emotional (feeling to be derived) statements of purpose (Ulwick 2005). *Outcomes* are structured as phrases taking the form: *direction / unit of measure / outcome desired* (Ulwick 2005), and *constraints* are formulated as statements that summarize an obstacle preventing product adoption.

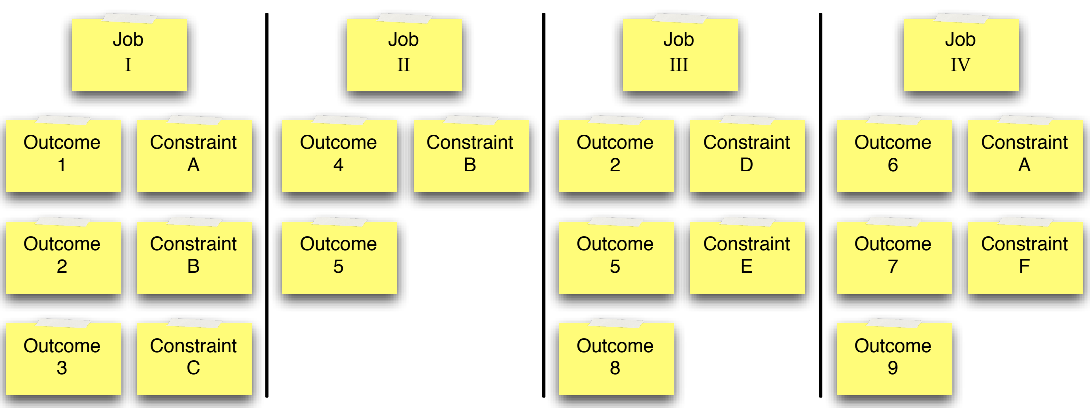


Figure 5.3 Example affinity sort used to group outcomes and constraints to jobs.

3. Sort the formulated job, outcome and constraint statements using an affinity diagram (Otto and Wood 2001) as illustrated in Figure 5.3.

Jobs should be used as the primary categories in the affinity sort. Outcomes and constraints should be grouped with their associated job categories.

4. Identify the overall customer goal from the jobs. The overall customer goal abstracts from the jobs to define the product opportunity and may be considered the rough *black box process*. *Events* abstract from the jobs.
5. Identify *flows* including the product being designed from the outcomes and constraints. Map each flow to the specific events where they are required. Add new ancillary tasks when flow-event mappings cannot be found.
6. Formulate an environment model from the customer inputs to define where the customer will use the product. This model defines the location where all flows will be derived as well as all operating conditions that the product will be subjected.
7. Formulate a black box process model from the overall customer goal identified in Step 4 and the flows identified in Step 5. This model defines the overall usage of the product being designed along with its associated energy, material and signal inputs and outputs.
8. Formulate an event model to consist of chains of events connected by flows of materials, energies, and signals that must be completed systematically to achieve the desired goal. The event sequence should begin with the initial product action and should be followed by all other discrete events including each of the actions, environ-

ments or situations where the product will be used over time. The product should be included as a flow through each event.

9. Decompose each individual event in the process model into a configuration model detailing the discrete changes to the product and any associated functional interactions with other flows in the event. Begin by generating chains of configurations. Once configuration chains capture changes from input to output, aggregate chains into complete configuration models.
10. Formulate a black box functional model to fulfill the circumstances abstracted by the process model. Functional models may be independently generated for each configuration change modeled in Step 9. Typically, however, functional models aggregate all operational aspects of a product design into a single functional model. This is a choice that is left to the design team. Flows required in the functional model should be derived from flows in the process model.
11. Decompose the black box functional model into functional models detailing the discrete changes to each flow within the boundaries of the product. Begin by generating chains of functions. Once function chains capture changes from input to output, aggregate chains into a complete functional model.

Once through the methodology, the models should be verified. Are each of the flows identified as outcomes captured in either the configuration or functional models? Are all operational aspects of the product modeled functionally? Once the models are verified, they may be used in either automated or manual conceptual design approaches where components and/or strategies may be mapped to functions and configurations. The aggregation of different components and/or strategies for each configuration and function should result in a complete set of strategies to fulfill the identified customer inputs for a product opportunity.

5.3 Example: Bosch CS20 Circular Saw

To demonstrate this methodology, consider the Bosch circular saw CS20 with Direct Connect. The Bosch CS20 is a professional-grade, circular saw (Bosch). In 2004, it was designated one of the most innovative products of the year by Popular Science (December 2004). Bosch's innovative redesign of the circular saw incorporates a number of innovative features including a dust blower to remove sawdust from the cut path and a direct connect system featuring a socket for an extension cord instead of a built in, fixed cord (Ulwick 2005).

Choosing a pre-existing product such as the Bosch CS20 Direct Connect as an example to the methodology simulates a product redesign procedure. As the majority of product design activity is redesign (Otto and Wood 2001; Ullman 2010), presentation of a redesign scenario is most relevant to this methodology's usage. Also, building on a proven example in literature of an innovative product demonstrates how this approach may promote innovative product design. However, the steps are equally valid for original design cases with the typical increased difficulty in executing the opportunity identification step.

Following the model generation portion of the methodology, Step 1 guides the design team to work with customers to identify inputs. In his text, *What Customer's Want*, Ulwick provides the jobs and outcomes identified by Bosch as important to the professional contractor. The identified jobs (taken from (Ulwick 2005)) are listed as Column 2 of Table 5.1, and are paired to a representative interpreted customer need from the affinity sort in Column 1.

Table 5.1 Job mappings to events for Bosch CS20.

Representative Interpreted Customer Need	Job (Ulwick 2005)	Event
Should accommodate a variety of different plans	Planning the cut	Plan Task
Many bevels and depths should be accommodated	Adjusting the saw	Adjust Product
Cutting should start without kick back	Starting the cut	Operate Product
Cut paths should be easy to follow while cutting	Operating the saw	Operate Product
Finishing of cuts should have minimal tear-out	Completing the cut	Operate Product
Maintenance should be quick and easy	Maintaining the saw	Maintain Product

Twenty key outcomes have also been identified and are available in (Ulwick 2005). Four of these outcomes are listed as Column 1 of Table 5.2. A key customer need for the Bosch CS20 is that the power supply must be reliable to provide electrical energy for long periods of time. This customer need is addressed with a constraint stating that the circular saw must be powered by 115V electrical energy as could be drawn by a standard portable generator.

Once the jobs, outcomes and constraints are known for a product opportunity, the overall goal is abstracted, following Step 2. For the Bosch CS20, the overall goal is to make a cut through a piece of solid material—likely lumber. The process would thus be to plan and make cut. Decomposing this process into events requires mapping jobs and ancillary tasks to events. These mappings are provided as Column 3 of Table 5.1. Each follows the verb-noun standard format of functional modeling.

Table 5.2 Outcome mappings to events and flows for the Bosch CS20.

Outcome (Ulwick 2005)	Event	Flows
Minimize the time it takes to make bevel adjustments	Adjust Product	Product, human material & energy, control & status signals
Minimize the likelihood that the saw will be stolen	Store Product	Product, human material & energy
Minimize the likelihood of the cut going off track	Operate Product	Product, human material & energy, solid material, electrical energy, control signal
Minimize the amount of dust/debris that is generated by the saw	Operate Product	Product, human material & energy, solid material, electrical energy, solid material (dust), solid material (debris)

Following Step 3, outcomes are used to identify the flows required for each of the events. Flows captured at the event level subsequently provide the flows required for its configuration and functional model. Through this mapping, each outcome is first mapped to an event identified in Step 2. Flows required for the completion of each event are subsequently identified from the outcome statement. These mappings are provided as Column 2 and 3 of Table 5.2.

Following empirical guidelines assists with making outcome-to-flow mappings. First, consider *who* or *what* will be performing the outcome.

Commonly, *who* is the customer that purchases the product and *what* is the product, but other tools or equipment may be required. Next, outcomes are abstracted by translating the natural language to the Functional Basis (Hirtz et al. 2002) lexicon. Finally, information (in the form of signal flows) related to each outcome is considered. Information can be the key to designing an intuitive product (Norman 2002), but it is not always explicit in the product requirements therefore some insight into how the product may be used is required.

These empirical guidelines may be applied to the outcomes listed in Table 5.2. For example, consider the outcome, "Minimize the time it takes to make bevel adjustments" (Ulwick 2005). *Who* will be the primary customer of the product, and using the Functional Basis, customer is represented as human material and energy. The product being designed would be *what*. Now, considering natural language, bevel adjustments, refer to cut angle information. This would be a control signal in the Functional Basis. Last, a flow related to how far the saw is adjusted would provide feedback. This is required for an intuitive product and is not explicit in the outcome. A status signal will refer to the set cut angle.

During this mapping, new jobs not explicitly known by the customer are likely be found. To address these jobs, new events are added to the process. For example, with the Bosch CS20, the outcome, "Minimize the likelihood that the saw will be stolen" (Ulwick 2005), does not clearly map to any of the previously identified jobs. To address this outcome, a new job, storing the saw, is added and mapped to an event; in this case, the new event, store product, is added.

Once the mappings are generated, integrated functional and process models are created. First, following Step 4, the environment is drawn. The environment defines the location where all flows will be derived along with all operating conditions for the product. For the Bosch CS20, this is the construction site where it will be used. In Figure 5.4, a double line represents the environment. Following Step 5, the black box process, Plan & Make Cut, is drawn. Single width line boxes represent the processes in Figure 5.4. All events and flows required in the event model come from the environment. The event chain, generated during Step 6, is also represented by single width line boxes in Figure 5.4. The five events, plan task, adjust product, operate product, maintain product and store product, are linked by flows identified from the outcomes during Step 3.

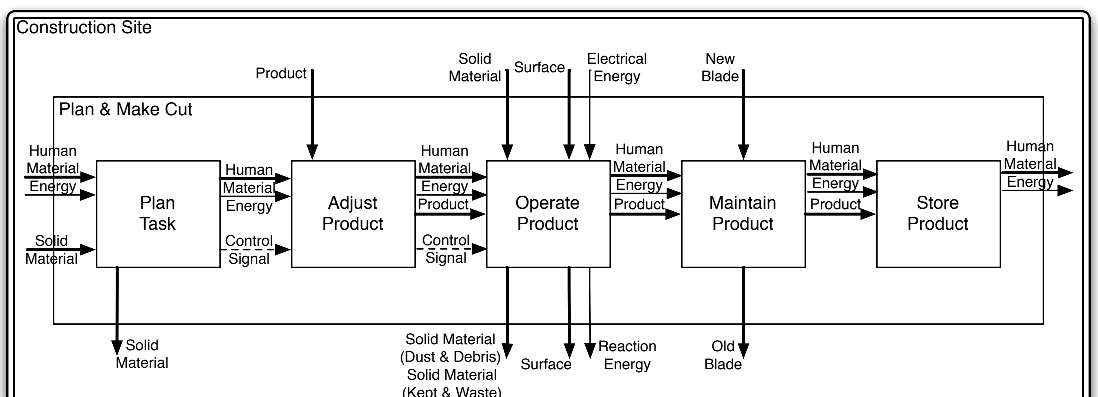


Figure 5.4 Environment, black box process and event model for the Bosch CS20.

Flow and event mappings identified during Step 3 are maintained when generating the event model. Flows are represented following the energy, material and signal modeling scheme where bold arrows represent materials, thin arrows represent energies and dash arrows represent signals. The product and all product constituents are modeled explicitly as flows in the process models to capture interactions and functional changes to the product as a whole. Flows required mid-event chain are drawn entering from the top of an event box, while the flows that are no longer needed by the event chain are drawn exiting the bottom. Flows that are drawn as crossing the boundary line for the black box process represent flow importation and exportation into the system.

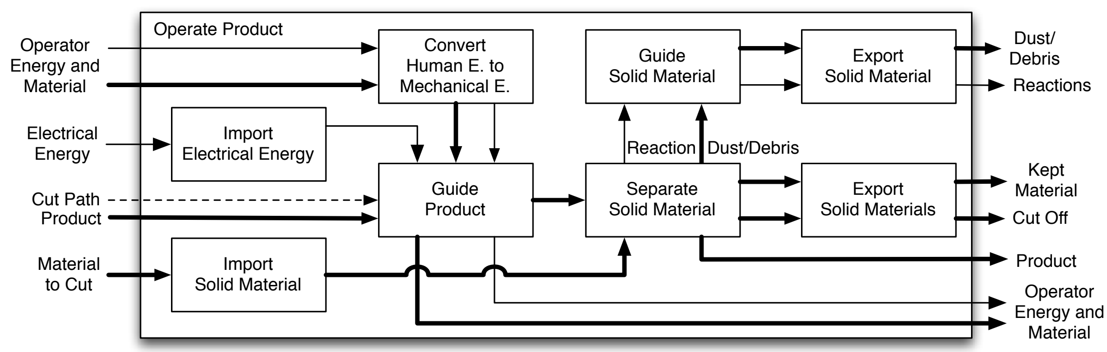


Figure 5.5 Configuration model of the Operate Product event.

Following Step 7, each individual event is decomposed into a configuration model. The configuration model describes specific customer-product interactions that result in discrete changes to the product as well as associated ancillary flow transformations required to arrive at the desired output flows for an event. Decomposition begins by first generating chains of configurations. Configuration chains are then aggregated into complete configuration models. The configuration model for the operate product event of the Bosch CS20 is Figure 5.5. Following primary / carrier notation (Nagel, Bohm et al. 2007), the operator (modeled as human energy and material) enters the configuration model and is converted to mechanical energy. The mechanical energy guides the product following the cut path (represented as a signal flow with a dashed arrow). The product separates the solid material resulting in dust and debris. The dust and debris are exported by an innovative feature of the Bosch CS20; a ducting system to redirect reactionary pneumatic energy generated during the separate solid material configuration. Also leaving the configuration model is the material to be kept, the cut off, the product and the operator.

The operator and product flows next enter the maintain product event and therefore are not exported in Figure 5.5. Only those flows no longer needed by the product are exported, and the converse is true for imported flows. Flows crossing the black box process boundary (Plan & Make Cut) in Figure 5.4 are imported and exported once in the configuration models. This use of the import and export functions is consistent with the definition provided by the Functional Basis, which states that import is used, “To bring in a flow from outside the system boundary” (Hirtz et al. 2002) and that export is used, “To send a flow outside the system boundary” (Hirtz et al. 2002).

Once configuration models are created for each event, functional models are generated for the product. Following Step 9, black box functional models are formulated. Black box functional models are stand alone functional models abstracting a high-level transformation intended for the product to complete. Black box and sub-functional models may be independently generated for each configuration change modeled in Step 6. Typically, however, a single functional model is created for the product. This functional model describes all operational aspects of a product as a single black box functional model and its decomposition. This is a choice that is left to the design team.

Once the black box functionality is extracted for the product, the flows required for the product to perform as intended are identified. The sum of the flows entering and exiting the product through the configuration models represent the minimum flows required by the product. The first pass black box model provided as Figure 5.6 begins with these flows from the process model. These flows in black box functional model are used when creating the functional decomposition following Step 9. The decomposition begins by detailing the discrete changes to each flow within the boundaries of the product. First chains of functions are generated for each flow capturing changes from input to output followed by the aggregation of function chains into a functional model.

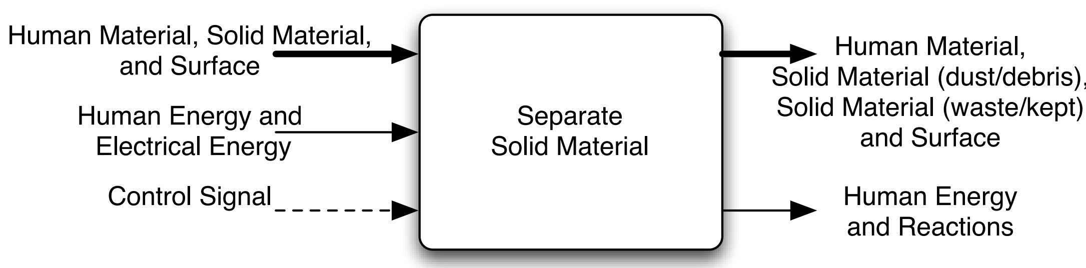


Figure 5.6 Black box functional model for the Bosch CS20.

The functional model for the Bosch CS20 provided as Figure 5.7 follows the transformation of each input flow modeled in Figure 5.6 to its desired output. In the model, 115V electrical energy is imported into the system. This input is known from the customer constraint defined in Step 1. The electrical energy is actuated from a control signal created by the operator. Once actuated, the electrical energy is used to create the mechanical energy necessary to rotate the saw blade. Human material guides the mechanical energy to the cutting location. Before the cutting action can be completed, the material to be cut must be imported into the system along with the surface to which the material is secured. Once the material is secured, it is guided by the operator and separated by the mechanical energy. The desired outputs, the surface, kept material and operator, now leave the system. However, due to inefficiencies, other outputs including reactionary outputs of pneumatic, acoustic, and thermal energy as well as waste materials of dust and debris also leave the system.

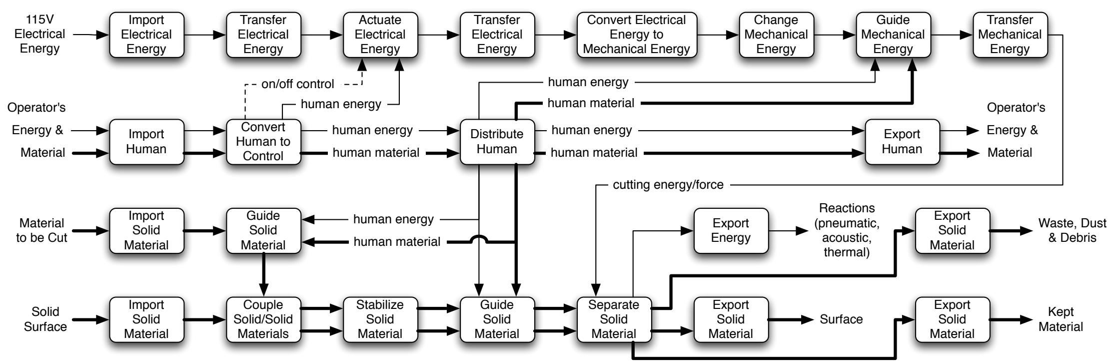


Figure 5.7 Functional model for the Bosch CS20.

Once the functional model is complete, all process models and functional models should be verified as discussed following the methodology. The design team should ask if each of the flows identified as outcomes are captured in either the configuration and functional models. Also, all operational aspects of the product for which design activities are required should be captured in a model. Detail should also be considered. Is more detail required; perhaps less?

Once the models are verified, the design team should move into the next phase of the engineering design process using function to seed concept generation. During concept generation, solution principles are paired to configurations and functionality. Chapter 8 discusses this process further.

5.4 Summary

Customer inputs from the Outcome-Driven Method have been explored as a starting point to function-based design. These inputs—jobs, outcomes and constraints—are based on the factors that influence customer product purchases (Ulwick 2005). By using these specific customer inputs to drive function-based design, problems related to the identification of proper customer needs, function and flow identification and voice-to-flow mapping can be mitigated. This research helps to formalize this leap from customer needs to function-based design.

Using Ulwick's customer inputs to drive process model generation leads to a starting point for flow identification and functional modeling. By defining first how a product will be used with the process layer, flow is captured outside the product boundaries by the circumstances rather than from customers—one of Christensen and Reynor's key criteria for developing innovative products (Christensen and Raynor 2003). From these circumstances, function and flow are identified based on how the customer and environment will interact with the product. These interactions and the required flows define the black box functional model and provide the key starting point to traditional function-based design (Otto and Wood 2001; Pahl et al. 2007). Functional modeling is then bound to and derived from the process layer creating truly integrated models capturing needs related not only to how the product will work, but also related to how the product and customer will interact.

Finally, moving from process into function provides a technique to work initially with broad usage ideas and to hone in on more specific product details systematically through abstraction levels. Each model layer captures more detail than the prior with the environment focusing first on where the product will be used. Then in the process layer, the black box defines what jobs the customer will be trying to complete when the product is employed. This job is broken down into more detail with events describing specific steps required to complete the customer's task. Once these steps are formulated, specific actions are considered in the generation of configuration models. Through these process layers, the product remains a flow with its functional changes recorded as configurations. These configurations and the related energy, material and signal flows define the black box functional model for the product, and provide the starting point for function-based design. The decomposition of a functional model follows with the creation of a function structure from the flow inputs and outputs captured in the configuration and compiled by the black box functional model.

CHAPTER 6 Process-based Failure Propagation and Impact Factors

Failure analysis is a fundamental part of the conceptual design process. Its integration with conceptual design can lower product failure rates and decrease required redesign efforts. The result of which is reduced design, and consequently, final product costs. Commonly, failure analysis approaches focus on the operability of the final product either through analysis of the product's functionality or its components.

Products fail, however, not only due to failures with internal components, but also because of failures in how they are used. To identify these types of failures, a failure analysis method based on process modeling is explored in this chapter. This process-based approach to failure analysis may be used for either conceptual design or systems analysis. During conceptual design, process-based failure propagation provides insight into error prone human-centric tasks. Designers may then account for these error prone human-centric tasks to reduce the likelihood of the final design embodiment failing. Alternatively, when applied to the usage of mature products or other established manual processes, the process-based failure propagation provides a systems analysis tool that can be used to identify potential failure points for process planning.

6.1 Background

A number of methods exist for identifying potential failures based on the operability of products. Of these techniques, Failure Modes and Effects Analysis (FMEA) is often considered the industry standard methodology for failure analysis. FMEA was originally developed from the failure modes and effects criticality analysis (FMECA) defined in MIL-STD-1629A (MIL-STD-1629A 1980; Stamatis 1995). Following MIL-STD-1629A, failure modes and their consequences are identified from functional and reliability block diagrams created for the system in question; this process is based on functional attributes of the system and does not necessarily require the components of the system to be known (MIL-STD-1629A 1980). Later, through an effort by Ford, Chrysler, General Motors and the Automotive Industry Action Group, FMEA in the automotive industry is standardized in the reference manual (Automotive Industry Action Group (AIAG) 1993). In practice, a FMEA may focus on the system (global functionality), design (components), process (manufacturing), service or software (Stamatis 1995). In all cases however, a FMEA is used to identify potential modes of failure, effects, consequences and actions. Figure 6.1 provides an example FMEA worksheet used during this process. Often in design, an FMEA is based on components making application difficult during conceptual design as this process tends to rely heavily on expert knowledge or system documentation not yet available to the analysts.

Failure Modes and Effects Analysis Worksheet						
System:________				Product:________		
#	Function Affected	Failure Mode	Failure Effect	Cause of Failure	Recommended Action	Action Taken
i						
Date: *Month Day, Year*				Performed by:________		
Company Name:________				Checked by:________		

Figure 6.1 Example form used to perform an FMEA.

Alternative FMEA-based approaches expand FMEA beyond these problematic function-component mappings. Russomanno et al. have proposed an Expert System for FMEA (XFMEA) (Russomanno et al. 1993). XFMEA automates FMEA activities and utilizes behavioral, functional and structural representations that allow failure analysis activities to be performed before component assignment has been performed. Advanced FMEA also tries to bring the failure analysis activities into the conceptual design phase by applying behavior models mapping control-based functionality to system components (Eubanks et al. 1997; Kmenta and Ishii 1998; Kmenta et al. 1999). This method identifies deviations from intended functionality that result from system failures. With Function Hazard Analysis (FHA), experts determine potential failures based on the behaviors of the system's functions to determine function-failure combinations. These function-failure mappings are based on system and subsystem functional decompositions (Wilkinson and Kelly 1998).

To identify the cascading or propagation of failures through a system Fault Tree Analysis (FTA) or Event Tree Analysis (ETA) may be used. FTA applies backward logic to develop a top-down chain of events which have the potential to lead back to a single negative event (Vesely and Goldberg 1981; Voland 2004; Blanchard and Fabrycky 2006). Events propagating to the negative event are modeled using Boolean logic, as illustrated in Figure 6.2 (a), to create chains of potential propagated failures. ETA, conversely, uses forward logic to investigate a single initiating event. From the single initiating event, probable failure events, which can occur in sequence, are analyzed to determine the likelihood of success or failure (Kumamoto and Henley 1996; Bedford and Cooke 2001). Figure 6.2 (b) illustrates an event tree diagram.

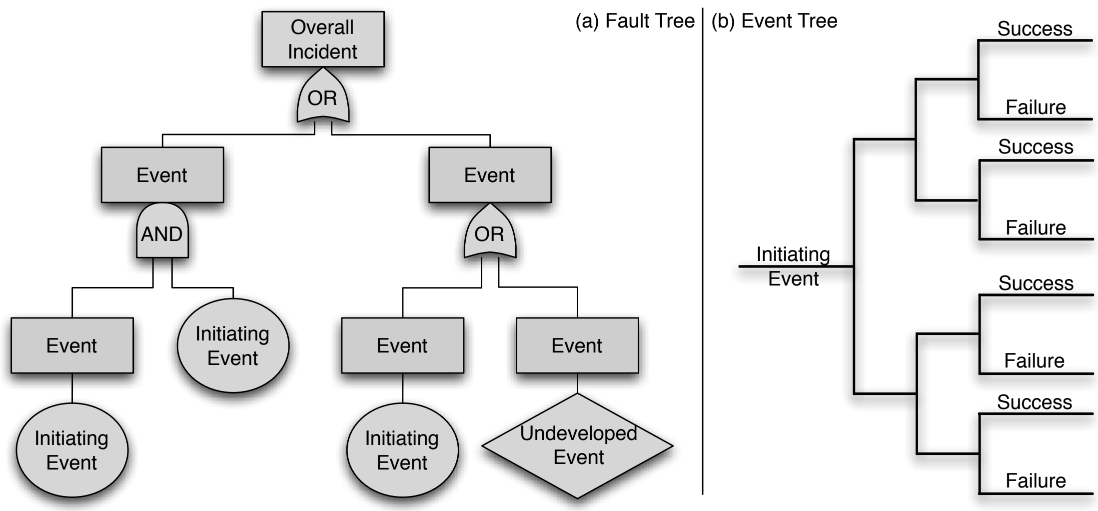


Figure 6.2 Example (a) fault tree diagram and (b) event tree diagram.

Both ETA and FTA can be performed on a system during conceptual design and can address internal and external failures making them suitable for both function and process based analysis techniques. Probabilistic Risk Analysis (PRA) combines the failure propagation techniques of FTA and ETA with failure effects identification techniques such as FMEA to answer three questions: what can go wrong, how severe will the failure be, and how likely is the failure to occur (Kumamoto and Henley 1996; Stamatelatos 2000; Bedford and Cooke 2001). PRA can be performed on internal and external initiating events through all phases of a system's life cycle. PRA, like FMEA, FTA and ETA, tends to rely on expert knowledge and does not typically employ structured modeling for system abstraction.

Approaches based on function, instead of components, have the advantage of being used during concept generation to evaluate components and solution principles as they are selected. For example, the Function Failure Design Method (FFDM) utilizes functional modeling with the Functional Basis (Hirtz et al. 2002) to apply failure analysis during the conceptual design phase and overcome the shortfalls of expert-based systems through the application of a knowledge-based repository of failure data (Stone, Tumer et al. 2005). Similarly, Function-based Failure Propagation and Functional Failure Identification and Propagation (FFIP) are proposed to analyze the failure propagation through function-flow system models utilizing the Functional Basis modeling lexicon (Hirtz et al. 2002). Function-based Failure Propagation identifies two failure modes that occur due to the propagation of failures along flows through a functional representation of a system (Krus and Grantham Lough 2007), and likelihoods are calculated based on information stored within a failure knowledge base. FFIP identifies functional failures and failure propagation in a system (Kurtoglu and Tumer "A Graph-Based Framework for Early Assessment of Functional Failures in Complex Systems 2007). In FFIP, failures are defined as negative events in the behavioral model. The behavioral model provides a mathematical representation of the potential components required of the system; the component representation is generated from the configuration flow graph. Failures may then be mapped back to the initial functional representation of the system and applied to similar functionality (Kurtoglu and Tumer "FFIP: A Framework for Early Assessment of Functional Failures in Complex Systems 2007). This differs from FMEA in that failures are mapped to potential behavior of the system instead of functionality or components. Jensen et al. (Jensen et al. 2009) introduce a logic-based reasoning approach to the overall framework of FFIP with Flow State Logic (FSL) to allow non-designed (or potential) energy, material and signal flow-based failure paths to be identified during failure analysis. With the addition of FSL, the FFIP method can be used to identify failures and failure propagation paths (both designed and non-designed) that are based on behavioral and functional representations of a system's design where exact behavior is unknown.

These approaches for identifying faults and their resultant failures deal primarily with the operability of the engineered system. To investigate processes, fault propagation graphs are proposed. Failure propagation graphs are based on structured hierarchical process and sub-process models and apply causal relationships to a set of possible failure modes (Padalkar et al. 1991). Fault propagation graphs applied to process models can be used to analyze failure propagation between the states, where states are defined as the different phases of an entire process. Causal relationships model the propagation of failures from a process to its sub-processes. Graph theory-based approaches may similarly be utilized to analyze the time and resource requirements of processes via PERT charts where processes are modeled as network diagrams with the nodes representing events and the arcs representing constraints (Marshall 1971), and to measure the effect of unexpected events modeled by PERT charts, Bowman presents a sensitivity analysis that can be used to estimate resultant probability distributions for program performance measures based on changes to project constraints (Bowman 2007).

6.2 Approach

Initial process-based failure analysis research explored existing manual processes following a system analysis perspective. This initial research explored manual processes of terrorist activities related to the development of Improvised Explosive Devices (IEDs) following the Red Hat of the adversary analysis approach. In a Red Hat analysis approach, Red Hat teams are established by an enterprise to challenge aspects of that very enterprise's plans, programs, assumptions, etc. In the case of counter-terrorism activities, such as the IED study, the Red Hat teams are trained and instructed to think as a terrorist adopting their views—both religious and cultural—through the study. It is this aspect of deliberate challenge that distinguishes red teaming from other management tools although the boundary is not a sharp one (Defense Science Board Task Force Sept 2003).

In the IED study, the Red Hat of the adversary approach was applied during the creation of process models (following Chapter 3, Section 3.2) that captured different manual processes related to IED incidents. Three tools grew out of the failure analysis of these process models: Propagated Failure Analysis, Process Failure Levels and System to Process Sensitivity. Propagated Failure Analysis (PFA) allows failures points identified through open-source research to be propagated through the process models. The impact of these failures is assessed using one of two impact factors. First, Process Failure Levels (PFL) is a qualitative impact assessment to classify potential scenarios resulting from identified failures, and second, System to Process Sensitivity (SPS) is a quantitative impact assessment of a single fault's propagation along flow paths at the configuration level of the process model.

6.3 Methodology

From the research performed on the IED incidents, the general method summarized by Figure 6.3 is extracted. First process models are generated following Step 1. The methodology to identify potential failure points is Steps 2-4. Then, to assess the impacts of the identified failures, Step 5 is followed for qualitative impacts and Steps 6-8 are followed for quantitative impacts. The specific details of each step are explained in the following sub-sections.

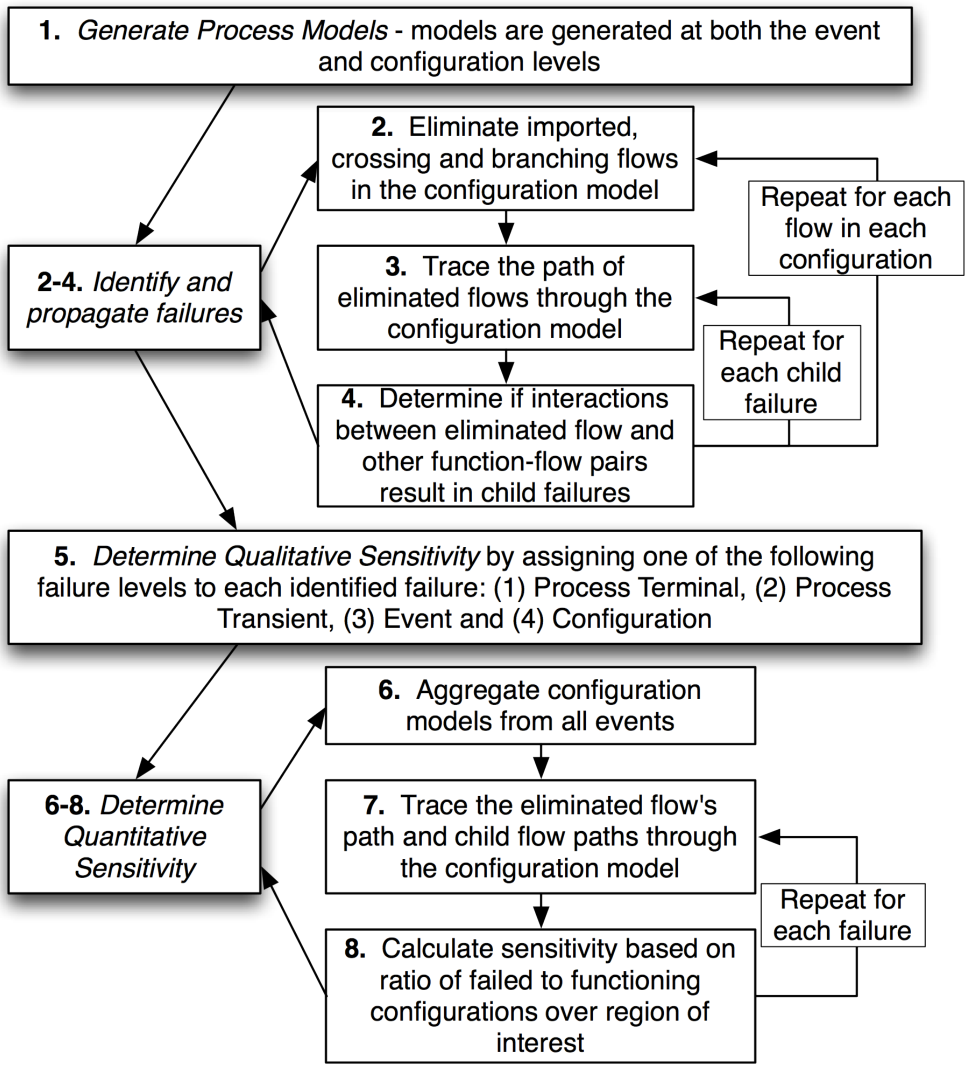


Figure 6.3 Methodology for analyzing propagated system configuration failures.

6.3.1 Propagated Failure Analysis (Steps 2 through 4)

Process models are first generated following the aforementioned methodology. If this method is being used as a standalone system analysis technique, then process models should be generated following Section 3.2 of

Chapter 3. Otherwise, if this methodology is being applied as a part of Conceptual Design activities, then integrated process and functional models should be generated following the methodology outlined in Chapter 5.

Once process models are generated for the manual processes, propagated failure analysis is applied to the models by considering the elimination of flows at three locations: (1) flow importation into a model, (2) flow branching in a model, and (3) flow merging in a model. These three flow elimination points are illustrated in Figure 6.4.

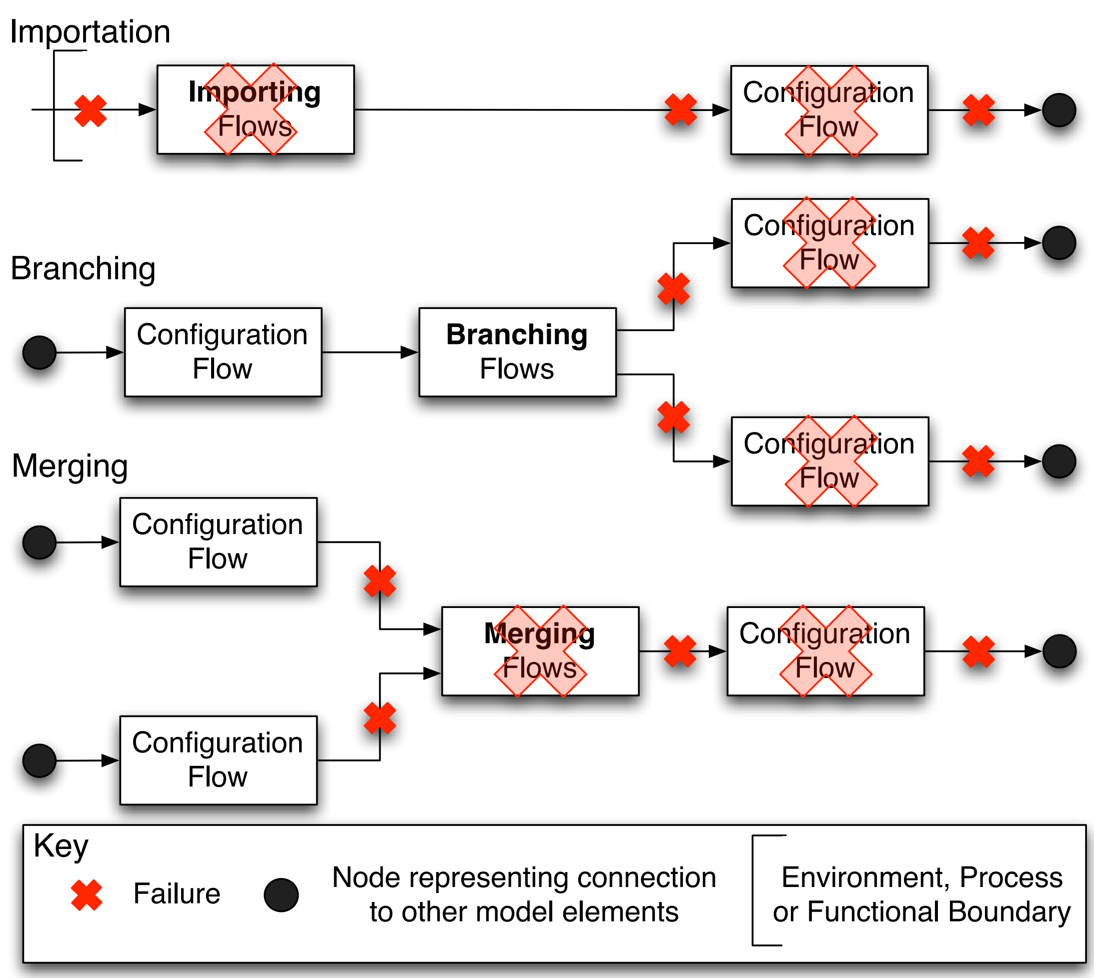


Figure 6.4 Illustration of flow importation, flow merging and flow branching in a configuration model.

PFA investigates how changes to these flows, vital to an overall process, affect the desired final outcome. Thus, PFA is based on each flow in a process model having a key role in the successful completion of the overall objective. If a configuration or flow is disrupted, the final desired outcome is also disrupted. The elimination of flows is propagated through configuration models; the eliminated flows are denoted with an X in the model. As failed flows are propagated, potential interactions, which result in other flows being eliminated, should be considered. Flows eliminated by other failed flows are termed child failures, and they too are traced through the system considering new interactions.

PFA utilizes two failure methods to show failure propagation through a process model. First is a "No Flow" failure (Krus and Grantham Lough 2007). A "No Flow" failure occurs when a configuration fails, thus stopping the output of the configuration from propagating further. This failure results in the termination of the flow on which the configuration acts allowing the failure to propagate along the flow path. The "No Flow" failure is demonstrated as the initiating failure in the diagram provided in Figure 6.5.

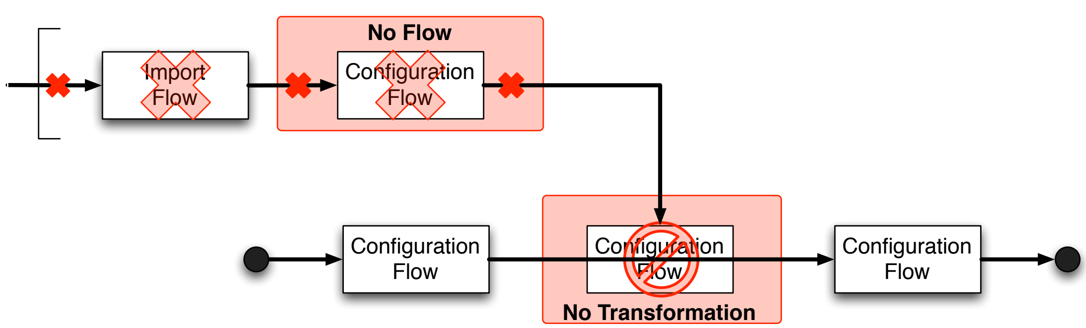


Figure 6.5 "No Flow" and "No Transformation" failure modes.

The second failure method is a “No Transformation” failure where a specific configuration fails, but flow through the configuration is not affected and the remainder of the chain continues to be operational. The configuration that fails by the “No Transformation” failure is marked by a circle with a slash as well as by the continuation of the flow arrow over the failed configuration block. The “No Transformation” failure is demonstrated as the second failure in the diagram provided in Figure 6.5.

6.3.2 Propagated Failure Level (Step 5)

At the completion of PFA, configuration models are marked with a number of potential failed configurations and flow paths. Each failure is now rated with a PFL considering whether the failure affects a single configuration, multiple configurations and consequently an entire event or process. PFL is a qualitative rating of the impact a failure has on the overall process based on projected scenarios for the continued operation of the process. PFL provides a way for an analyst to denote those failures having the potential to be more devastating to the system as a whole.

From the PFA performed on the modeled IED processes, four distinct types of failures were identified which has lead to four PFL ratings: (1) **Process terminal failure** is a failure mode that ends a process completely and in such a way as to render the process permanently terminated. (2) **Process transient failure** is a failure mode that ends a current instantiation of a process, but the process can be restarted at a future time. (3) **Event failure** is a failure mode that occurs within an event of a process but the overall process remains operational. (4) **Configuration failure** is a failure mode that occurs when a single configuration change within an event can no longer occur.

Since failures propagate through an entire process, a minor failure during one event might result in a more significant failure at a later event. Thus, a single failure might result in more than one PFL. Also, since the PFL is a tool to assess the impact of various scenarios for how the remainder of a process will operate, different scenarios are often assigned a different PFL.

6.3.3 System to Process Sensitivity (Steps 6 through 8)

Last, the SPS is calculated for each failure. SPS provides a percentage of the flow paths failed in the configuration model due to single initiating failure propagated over the sum of all configurations within a desired range of an aggregated configuration model. The sensitivity metric requires that configuration level models from all events in a process be compiled into a single aggregated configuration model. To create this aggregated configuration model, each configuration model is connected following the flows that connect the event from which they were decomposed. For example, in Figure 6.6 configurations, C1 and C2, transforming Flow A in Event 1 would aggregate with the configurations C3 and C4 in Event 2 that also transform Flow A.

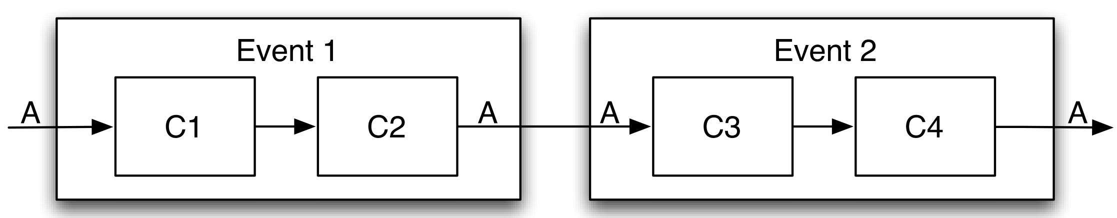


Figure 6.6 Example aggregation of configurations.

The failures identified during the PFA are then traced over the region of interest through the aggregated model. A summation of failed configurations, C_f, is used to generate a ratio, termed Impact Factor (IF), relating the failed configurations to the sum of all configurations, C_t, in the region of interest. The impact factor is calculated following Equation (6.1).

$$IF = R_i = {C_f}/{C_t} \tag{6.1}$$

SPS provides a gauge to the configurations that will be failed due to a single initiating failure. For instance, if the sensitivity of the entire system to a single failure is desired, then each configuration model from all events must be compiled into a single aggregated configuration flow model. Each of the failures identified during PFA are then traced through this aggregated configuration model. SPS provides a numerical way to quantify the affect of a failure over a specific range in a configuration model and are in no way meant to represent the severity of a failure. They are instead meant as a numerical guide to compare multiple failure modes of a specific range of configurations in a process model.

Sensitivity analysis based on a statistical stack-up algorithm (Ullman 2010) provides insight into the relationship between identified failures in a system. Each failure is compared as a percentage of the total failure profile. The resultant sum of all failure contributions equals 100% of the total failure profile. Using a statistical stack-up algorithm provides insight into which failures contribute most to the failure profile for a system. To standardize the relationships (R_i), the overall deviation is calculated using the Root Sum Square (RSS) method; this equation is provided as (6.2).

$$R^2 = R_1^2 + R_2^2 + R_3^2 + \ldots + R_i^2 \tag{6.2}$$

The sensitivity contribution (S_i) of each failure to the overall failure profile is calculated following Equation (6.3) with the division of the square of the single failure relationship by the square of the relationship deviation. The final sum of the overall failure profiles sums to unity following Equation (6.4).

$$S_i = {R_i^2}/{R^2} \tag{6.3}$$

$$1 = S_1 + S_2 + S_3 + \ldots + S_i \tag{6.4}$$

The stack-up algorithm scales the sensitivity contributions to unity ensuring that failures over the same region of interest may be compared, while the use of configuration models will allow different potential product configurations to be compared to identify alternative product interactions that are more or less sensitive to failures.

6.4 Example: An Improvised Explosive Device Incident

Consider one of these cases investigated for the JIEDDO (Joint IED Defense Organization) where the prelude, development and execution of an IED incident are performed by a managed and highly structured terrorist cell. This example represents just one of multiple terrorist cell structures identified through open-source research conducted during this study. Before discussing the example, it is important to understand that the process models described in Section 6.4.1 would differ for different terrorist cell structures. Each different case would result in unique process decompositions with different failure analysis results.

6.4.1 Process Model Generation (Step 1)

For the managed and highly structured terrorist cell example, open-source research performed through literature reviews, homeland security documentation and Internet searches lead to the identification of twelve key events comprising an IED incident. These events are listed in Table 6.1.

Table 6.1 Twelve key events identified through open-source research comprising an improvised explosive device incident.

IED Events		
0. Manage & Recruit Terrorists	4. Gather Intelligence	8. Deploy IED
1. Train Terrorists	5. Acquire Materials	9. Initiate Attack
2. Wait Terrorists	6. Build IED	10. Detonate IED
3. Determine Target	7. Store IED	11. Evade Detection

These twelve events are aggregated into an event model to represent the IED incident. The event model, included as Figure C.1 in Appendix C, captures the three phases of an IED incident—prelude, development and execution—as well as the distribution of specialized tasks through the managed structure.

The event model begins with the prelude phase. Prelude consists of events for recruiting and managing the terrorist cell and collecting target information. Recruitment agents bring potential terrorists together. Newly recruited terrorists are provided with training materials and trained for specific tasks, and once trained, the terrorists wait for their assignment to specialized tasks. For clarity, each task is modeled as a separate branch in the event model. The first branch completes the prelude phase of an IED incident. During these events, surveyors identify potential targets and gather intelligence. The process of gathering intelligence may continue over a period of days or weeks, which is represented with the bold face arrow leaving and reentering Event #4, Gather Intelligence. Gathered target information is returned to the managers and recruiters in Event #0, Manage & Recruit Terrorists. This information feedback is represented with the dashed arrow, Target Information. Managers determine the type of IED to develop, build and deploy. This information is fed forward to the terrorists assigned each specific task.

The remaining two phases consist of development and execution. First during development, a supplier acquires the appropriate materials for the IED, a bomb maker constructs the IED and the completed IED is stored. In the final phase, execution, the emplacer retrieves the IED from its storage location and deploys it; and if required, a triggerman targets the target and detonates the IED before trying to evade detection.

Once the specific events are aggregated, each event is decomposed into a configuration model. Consider the event, Gather Intelligence, as an example. During the Gather Intelligence event, a surveyor monitors the targets' habits, interprets the habits, collects information and then supplies the information to the managers. The configuration model, provided as Figure 6.7, represents the surveyor entering the event as two flows, one thin, representing energy and one bold, representing material. The surveyor is transferred and guided by the notion of a selected target, represented as a dashed arrow, to detect the potential target. Intel, also represented as a dashed arrow, is processed, collected and stored by the surveyor. A feedback loop, represented as the bold and thin arrows of the surveyor, signifies the repetition of the surveyor watching the actions of interest before the intelligence is supplied to the managers. Once the surveyor has completed the assigned task, the surveyor is exported from the gather intelligence event.

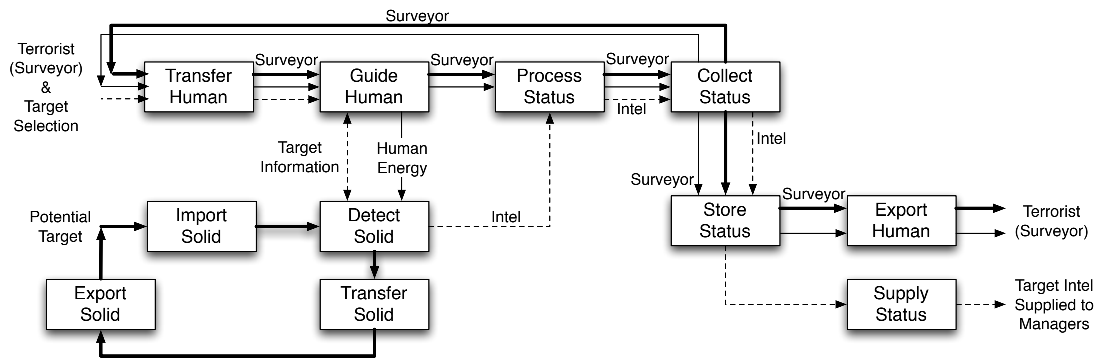


Figure 6.7 Configuration model for the Gather Intelligence event.

Like with other configuration models, each event in the IED incident is decomposed similarly to the example configuration model for Event #4, Gather Intelligence, by considering each of the operations that must occur on the flows entering the event. These operations are formulated as chains of configurations and aggregated into a complete configuration model representing interactions occurring between the operator and the system.

6.4.2 Propagated Failure Analysis (Steps 2 through 4)

The configuration models are now used as a starting point for PFA. Based on the open-source research for the IED incident, failure points are considered at three locations: (1) flow importation into a model, (2) flow converging in a model, and (3) flow branching in a model. The eliminated flows are propagated through configuration models.

For example, if the surveyor is stopped while monitoring the actions of the potential target, then the target cannot be detected, intelligence cannot be collected, and there is no intelligence to transfer to the leaders of the terrorist cell. The initiating failure, stopping the surveyor, is represented by a shaded X on the flows and the configuration, transfer human, in Figure 6.8, and its failure propagation is represented by an unshaded X. For tractability, large Xs have been placed on configurations and small Xs have been placed on flows. The failure results in "No Flow" for the surveyor since the surveyor is no longer present to perform the assigned task; however, since the potential target is still carrying out its standard routines and actions, a "No Transformation" failure occurs with the detect solid configuration. The circle with a slash over the failed configuration, detect solid, as well as the extension of the potential target flow arrow represents the "No Transformation" failure. An unshaded X denotes the child failure occurring with the flow of intelligence at the configuration, detect solid.

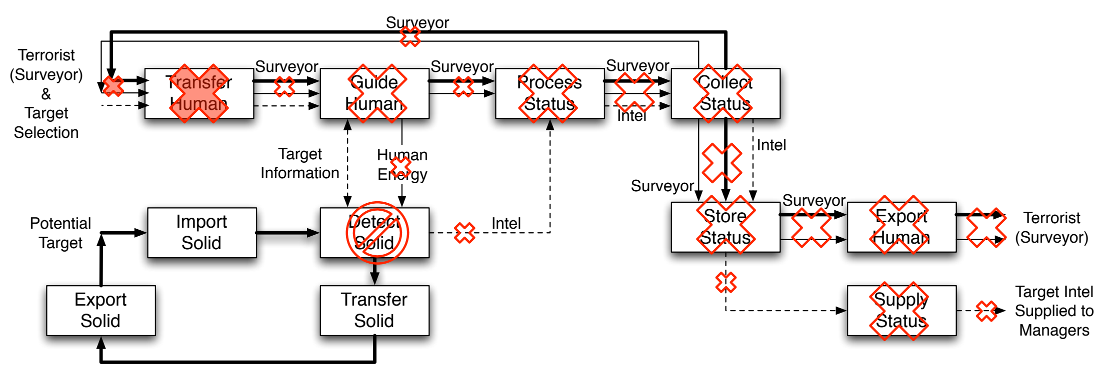


Figure 6.8 Configuration model with a failure on the surveyor flow.

If, however, the observability of the potential target were to be reduced such that the surveyor were no longer able to reliably gather intelligence by monitoring the potential target's actions, then an effective failure would be created on the potential target flow. This potential failure is modeled in Figure 6.9.

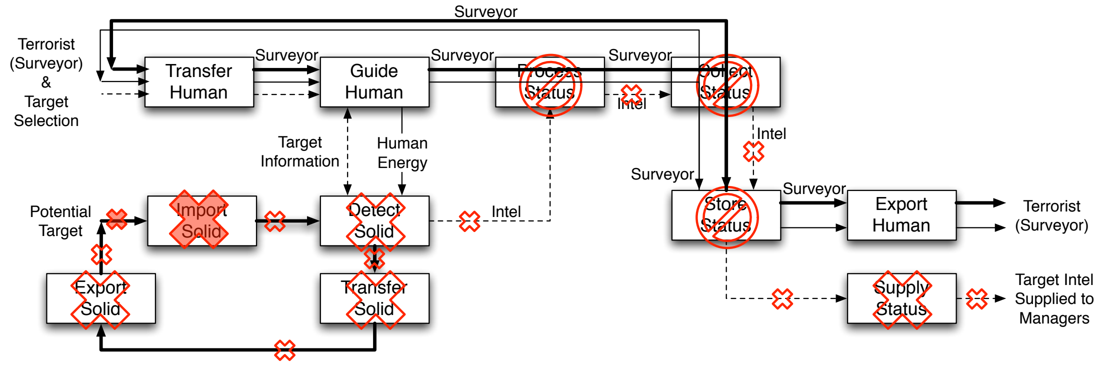


Figure 6.9 Configuration model with a failure on the potential target flow.

With this failure on the potential target flow, the initially failed flow and configuration block are, again, represented with a shaded X, and the propagation of the failure along the flow paths is represented with an unshaded X. This failure scenario results in a "No Flow" for the configurations associated with the potential target—import solid, detect solid, transfer solid and export solid all are failed since none of the configurations are able to be performed following the initiating failure. Configurations dealing with processing, collection and storage of intelligence would, however, each have "No Transformation" failure since the surveyor is still present in the model and is still able to perform the assigned configurations. A circle with a slash is, again, used to represent the "No Transformation" failure, and the surveyor flow passes over each of the failed configurations.

6.4.3 Propagated Failure Level (Step 5)

A number of different scenarios can be considered for how the remainder of the process will be performed with a failed surveyor flow. Consider, first, that the surveyor is irreplaceable, and the managers of the terrorist cell refuse to take action without proper intelligence; failing the surveyor would then be a process terminal failure. If, however, the surveyor can be replaced, which is unfortunately the likely scenario since the process is continuous and there are numerous recruits, then a process transient failure occurs and the process will restart once a new surveyor is in place. It is probable, however, for the managers to act on inadequate intelligence, miscalculate the actions of the target and proceed with a terrorist act that has a reduced likelihood of success; in this case, an event failure has occurred.

Second consider a scenario where the counter-terrorism activity is to reduce repeatability of the potential target shown in Figure 6.9. While this scenario does not actually cause a negative result to the target flow, it does result in an effective failure where the repeatability of the potential target (i.e. convoys, troops, etc. vary their routes and transit times) is reduced. Tracking would be more difficult for the terrorist cell making troops movements more difficult to record. This failure results in configuration failure on the intelligence flow where it is still collected, but predictability is reduced.

6.4.4 System to Process Sensitivity (Steps 6 through 8)

Last, the SPS is calculated for each failure. SPS provides a quantitative impact for each failure over a region of interest. Consider again the two failures in the Gather Intelligence event: (1) stopping the surveyor from monitoring our troops and convoys (failure on surveyor flow) and (2) reducing the predictability of our troop and convoy movements (failure on target flow). Assuming that the entire process is the region of interest, then the configuration models must be combined, and the failure propagated through the entire model. To generate the combined configuration model each events configuration model is aggregated based on the flow connectivity originally modeled by the event model (available as Figure C.1 in Appendix C). The resulting combined configuration model contains 105 configurations. Table 6.2 provides the calculated impact factors for the two failures.

Table 6.2 System to process sensitivity calculations for two potential counter-terrorism activities.

Failure	Total Failed Configs	Total Configs	Raw Impact Factor
(1) Stopping the surveyor from monitoring troops and convoys	20	105	$\frac{20}{105} = 0.190$
(2) Reducing the predictability of troop and convoy movements	12	105	$\frac{12}{105} = 0.114$

The SPS determined for the failures, stopping the surveyor from monitoring our troops and convoys and reducing the predictability of our troop and convoy movements, shows insight into the differences between the SPS and the PFL. Stopping the consistency of the troop and convoy movements has an impact factor of 0.114 indicating that 11.4% of the flow paths through the combined model will fail. This failure was assigned a configuration level PFL since the event can still be performed with limited reliability once the failure occurs, which logically should indicate that the failure is limited to affect only configuration changes within Event #4 where the failure initiates. However, had the configuration level failure been isolated to Event #4 then it would have a sensitivity of less than 8.3% for 1 event out of 12 total events; this was not the case. Instead, the failure follows the target information signal flow illustrated in Figure C.1 propagating back to Event #0, Manage & Recruit Terrorists. Propagating the failure back to Event #0 affects how the decision makers of the terrorist cell manage the cell and plan IED incidents.

A unique PFL was assigned to a number of different failures for the surveyor flow depending on the leadership of the terrorist cell. Stopping the surveyor flow, if assigned an event level failure to indicate that the managers operate on reduced intelligence, has the potential to fail 2 events out of the 12 total or 16.6% of the process. This failure to the surveyor flow is consistent with the 0.190 impact factor where 19% of the flow paths will fail.

Other failures identified for the managed IED incident during the study are provided in Figure 6.10 in the form of a Pareto chart (Blanchard and Fabrycky 2006)—the benefit of which is to identify, in rank order, the potential failures that will have the greatest effect on the system under study. Three formats are provided for the data: (1) the raw impact factor as a ratio of faulted configurations to total configurations in the region of interest, (2) the ratios scaled with a statistical stack-up algorithm, and (3) a cumulative percentage illustrating the contribution of each failure to the total deviation for the failure profile of the system.

When the entire failure profile is scaled using the stack-up algorithm and plotted in a Pareto Chart, failure trends may be investigated. For instance, when investigating the failure profile for a managed IED incident, it becomes clear that the first four failures—(1) stopping managers and recruiters, (2) stopping potential terrorists, (3) stopping terrorists from and while gathering supplies and (4) stopping the availability of supplies—have the the largest impact to the overall system. These failures fall generally within the Pareto 80-20 rule where 80% of the total impact is generated by 20% of the elements in the set. In this case, these four failures represent 25% of the total failure profile and 84% of the total impact. The final seven failures—stopping the predictability of target movements through stopping the terrorist after detonation—generally contribute less that 1% of the total impact to the failure profile indicating that the benefit of causing these failures from a counter-terrorism perspective is minimal.

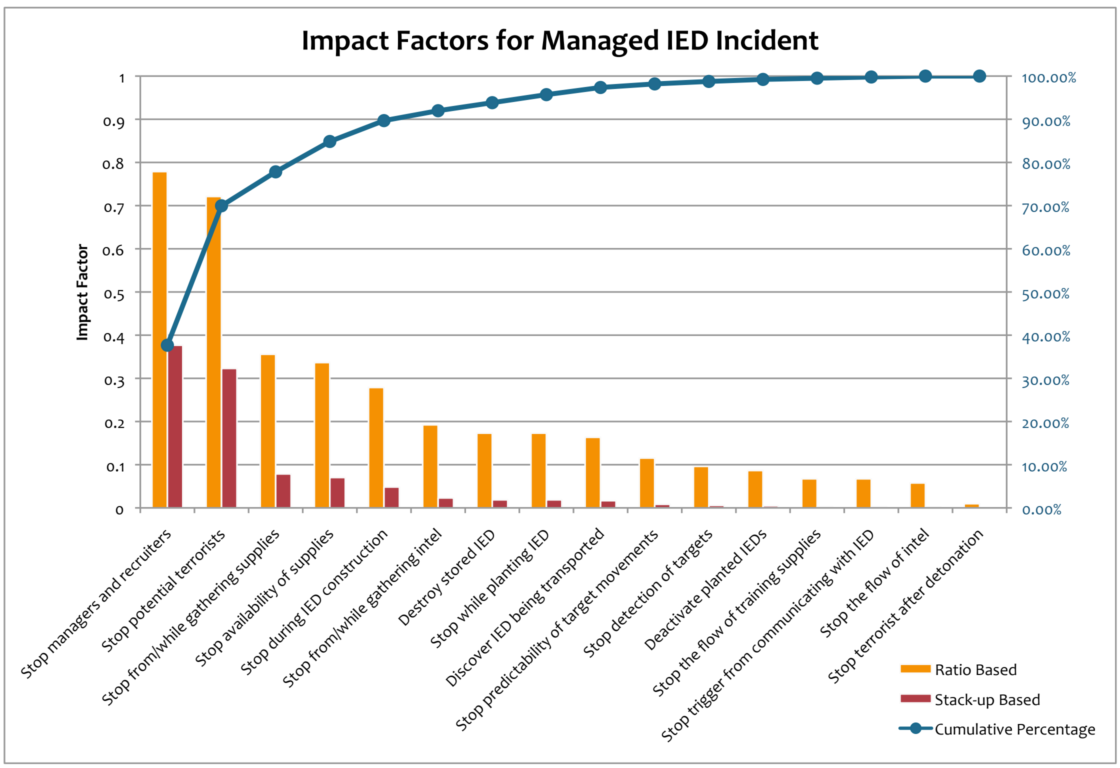


Figure 6.10 Impact factors for a managed IED incident shown as the raw ratio, scaled with a statistical stack-up algorithm and compared with a cumulative percentage.

Also, from a philosophical point of view, it is important to note that while failures were rated with a PFL of process terminal, there are no failures with an impact factor of 1. The highest impact factor is, in fact, 0.778. This discrepancy is a result of the SPS calculation. Since, SPS is an indicator of the configurations failed along flow paths, flows that are still available as inputs to the system, but are no longer utilized, are considered as resources still available. These available configurations are not counted as a part of the failed configurations making SPS values of 100% unlikely.

6.5 Summary

The preceding example, based on IED deterrence, demonstrates the application of PFA, PFL and SPS toward understanding failure propagation in a system analysis application. In the area of engineering design these tools are applied similarly to investigate potential failure paths related to how the customer uses the product. Using the configuration model to understand potential customer-product interactions provides a methodology to investigate unexpected customer actions during the conceptual design process when functional and process details are initially being distilled from a customer needs set. This analysis may be used to verify redundancy in critical systems or for removing redundancy in disposable systems. To illustrate, consider the Bosch CS20 from Chapter 5 as a simple example. A process model generated of a circular saw may indicate that when cutting large surfaces, saw dust tends to accumulate over the cut line, making it difficult for the user to follow the desired path; this might be redesigned to duct excess air generated by the motor toward the cut path ensuring that dust does not have an opportunity to accumulate. (See Chapter 8 for a more detailed description of this example.) Design applications of these failure analysis tools are discussed in Chapters 8 and 9.

CHAPTER 7 Automating a Human-centric Process

Customer actions associated with the use of products provides a key starting point for the design of automated systems. These human-centric processes describe the activities of customers, and can provide valuable insight into the functionality required of an automated system. Automated systems such as these tend to be mechatronic—blending mechanical, electrical, computer and control systems into a single synergistic product. There is no rule, however, stating that they must be mechatronic, therefore, for generality, they will be termed automated systems.

Whether the product is truly mechatronic or automated, functional and process modeling lend themselves well to this type of multi-disciplinary design process due to the aggregated nature of their models. Function chains detailing the transformations required of each flow are often generated independently, but in a final model, they are aggregated to illustrate how each flow must interact to affect the desired transformations and bring about the desired customer needs. In this chapter, functional and process models will be used to explore how customers interact with a product they currently use in a human-centric process and how those products operate to meet their needs. Elements from the configuration models (generated from the existing human-centric process) will be strategically selected and aggregated with the core functionality of the product (generated from the existing product). This new combined model may then be used during concept generation to develop potential solutions for automated systems to replace the customer's current manual product operation.

7.1 Approach

As a methodological statement, both functional and process models are prescribed to be generated in conjunction with the collection of customer inputs to design products that assist customers with the completion of manual processes. Potential automated subsystems to assist with each manual process are then synthesized—in the form of a functional model—for each manual process based upon the functional and process models. Functional and process models are aggregated into a new functional model and used during concept generation to identify solution principles to replace human-centric actions. During these research activities four empirical guidelines are postulated:

Guideline 1: When creating a conceptual functional model for an automated solution, functionality can be derived from the customer's current process and the products currently used as a part of the process. From the functional models developed for the products, function chains abstracting the core functionality of the products can be extracted. Core functionality is the function-flow pairs that are essential to the operational event of the product. Those function-flow pairs not considered core functionality are those related to events such as maintenance or storage. From the process models, customer actions indicate sensing, operability and mobility requirements for an automated solution. Using the Functional Basis and its associated grammars provided in Appendices A and B assists with the aggregation of functional and process chains.

Guideline 2: Human-based interactions with the product require little change when considered for the functional model of the automation solution since the functional model is developed at a conceptual level. *Human energy* and *human material*—both secondary Functional Basis terms—may be rewritten as their primary level terms, *energy* and *material*, to represent an unknown source in the automated solution.

Guideline 3: Process and functional models of the manual process often reveal an array of human senses such as vision with eyes and tactility with skin that obviously cannot exist in their natural form in an automation solution and must be replaced with engineered solutions. Sensors, whether in humans or automated devices, also require processing. Sensors and their processors, however, when modeled functionally with the Functional Basis, have the same functionality whether they are solved via a natural solution or an engineered solution and only require an energy source change. Again, the grammars provided in Appendices A and B can assist with this conversion and their aggregation.

Guideline 4: Customer inputs may change with full automation of manual processes. Formal methods for determining customer inputs should be followed in conjunction with the process and functional decompositions of the customer's current process to ensure that all customer inputs are identified. The inputs from Ulwick's Outcome-driven Method (Ulwick 2002; Ulwick 2005) should be used to initiate functional and process modeling following the techniques of Chapter 5.

7.2 Methodology

It is important to fully explore and understand the customer's current actions, their jobs and outcomes, and the products currently being used. Exploring the processes and functionality related to the existing manual processes should reveal the majority of the needs of the target customers. Thus, process and functional modeling are employed to explore the manual process and any products currently used by the customer. Also, to ensure that shortcomings with the existing product do not become shortcomings in the new product, the customer needs are still identified. The customer needs are correlated both to the customer's expected outcomes and to the current process. Outcomes not addressed in the current product are addressed through the addition of new product functionality.

From the collected functional and process information, a black box model is generated for a product to automate the manual process. The black box model considers the overall functionality and all input/output flows. Its decomposition into a functional model details the transformations required for all input and output flows of the desired automated system. Once the functional model is generated, it is verified with the collected customer needs and automation objectives to ensure that all requirements are met.

The following six step methodology more formally describes this process:

1. Determine the automation objectives. Discuss the current process with the customers. Discuss the products currently being used. What are the shortcomings? What are the benefits? Perform an ethnographic study (i.e., monitor the customer while they carry out the current process).
2. Develop models of the current process and any products currently being used during the process following the procedure outlined in

Chapter 3. Use the Functional Basis (Hirtz et al. 2002) for terminology to ensure consistency between both the functional and process models.

3. Correlate customer needs and automation objectives to the process and functional models. If discussions with the customer reveal needs are not met in either model, use the unmet customer needs to determine additional flows to address the unmet needs.
4. Develop a black box model for the automation solution considering the high-level functionality and the input/output flows from both the process models and functional models of the existing manual product as well as the flows mapped to unmet customer needs in Step 3.
5. Develop a conceptual functional model for the automation solution by:
    a. Extracting the core functionalities from the functional models of the products currently used in the manual process,
    b. Converting human interactions such as mobility, actuation, sensing, operational energy, et cetera from the process models into non-energy specific functional equivalents,
    c. Aggregating the core functionalities with the non-energy specific functional equivalents for the process-based product interactions,
    d. Developing and aggregating function chains for the flows mapped to unmet customer needs during Step 3.

6. Verify that all of the customer needs are addressed by functionality within the final functional model. If they are not all met, return to Step 3 to identify and borrow the functionality in original current processes. If the functionality is not in the original process or functional models, address the customer needs by correlating the customer needs to outcomes and the outcomes to functions and flows. Aggregate the additional functionality into the newly synthesized functional model, and repeat Step 6.

Following the application of the above methodology, the designer has a functional model representing the required operations for a product to automate or assist a customer with a previously manual process. At this point, the designer is in the conceptualization phase of engineering design where the functional model is a key step that ensures customer needs are fully captured and represented in potential solution principles (Otto and Wood 2001; Pahl et al. 2007).

7.3 Example: Manual Can Opener

When developing an automation solution to replace or assist with a manual process, it is important to fully understand the manual processes and the product being replaced with the new design. To that end, consider, the manual process of opening a canned food item. Step 1 of the methodology directs the designer to identify an automation objective. Let us assume that manual can openers available on the market do not provide adequate mechanical advantage to allow operators to easily remove the lid from a can.

Thus, the automation opportunity is with the mechanical advantage required to remove the lid from a canned food item.

Once the automation opportunity is identified, Step 2 directs the creation of both process models and functional models for the current manual process and the product used by the customer. Consider first the functional model of manual can opener shown in Figure 7.1. Since the generation of the functional model for the can opener is created from the physical product, the flows may be identified from the physical product and their transformations (functionality) may be identified by *being the flow* (Otto and Wood 2001).

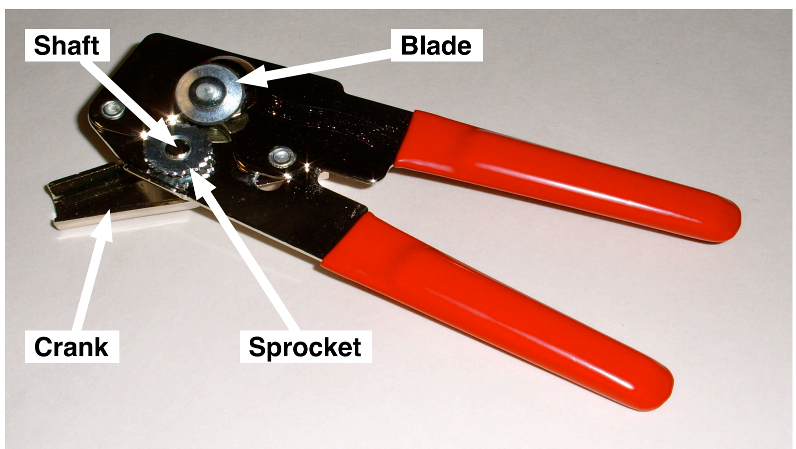


Figure 7.1 Manual can opener.

To generate a black box functional model for the manual can opener, the high-level functionality of the product is first identified. For the can opener, the objective is to remove a can lid. In the Functional Basis, remove pairs with *separate*, and the *can lid* is a solid material; thus the black box functionality is *separate solid material*. Flows required to perform the overall functionality are next identified. For the can opener, the input flows can be identified by watching the product being used. These flows include: the operator, the operator's energy (since it is manually powered, controlled, etc.), an unopened can and status on the system. These flows will be transformed by the product to arrive at output flows that include: the opened can, its lid, the operator, resultant reactionary energies and a change in the status. The complete black box functional model for the manual can opener is Figure 7.2.

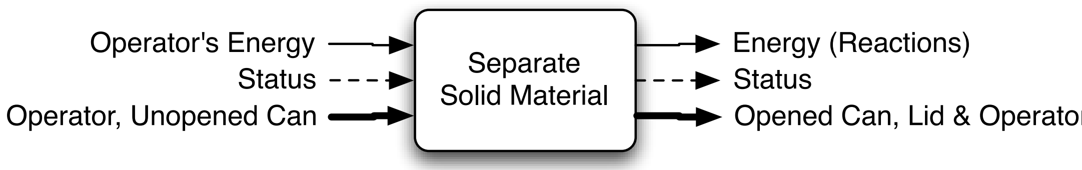


Figure 7.2 Black box model of a manual can opener.

The functional model of the manual can opener, decomposes the functional black box as chains aggregated into a complete functional model. The manual can opener functional model in Figure 7.3 follows the operator, the operator's energy and the unopened can through the product describing the transformations of each flow required to deliver the desired outcome. The operator's energy is first imported into the can opener at the crank as human energy. The human energy is then converted to mechanical energy through the act of rotating the crank. A shaft transfers the mechanical energy to the sprockets, which guide the can (modeled as solid material) along a rotating blade removing the can's lid (also modeled as solid material). Once the can's lid has been removed, the desired operation is complete, and all flows are exported from the product. Flows imported and exported from the system match those flows identified during the creation of the functional black box for the manual can opener.

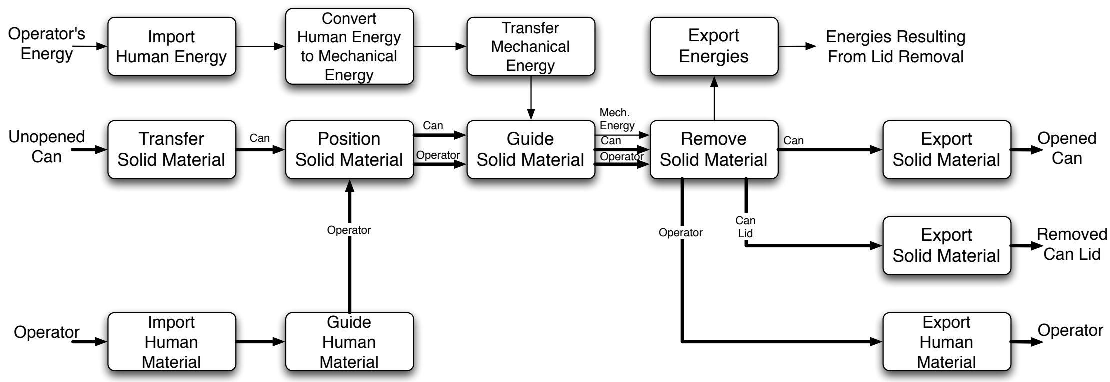


Figure 7.3 Functional model of a manual can opener.

Process models are also generated to describe the manual can opener by considering the actions taken by the customer as they use the can opener to remove a can lid. Let us consider two potential jobs during its ownership, can opener storage and can lid removal. During these jobs, the customer expects the can opener to easily configure for storage as well as for can lid removal. Since the can opener is manual, it must have a mechanical advantage to afford operation to a wide range of operators, and finally, since the diameter of a can is non-standard, the operation must be independent of the diameter of a can.

Knowledge of the customers' jobs is used to compile the individual events expected during the life of the can opener. The job concerning can opener storage will be *store product*, while the job concerning the removal of a can lid will be *operate product*. Flows necessary for the process may be identified through observation and include the can, can opener, operator and the operator's energy. The event model shown in Figure 7.4 links the identified events to their associated flows. Only the material flow of the can opener connects the events. The material and energy flows for the operator are present in both events, but are discontinuous between the storage and operation events to represent that these flows do not necessarily represent the same operator and may occur at separate discrete instances in time.

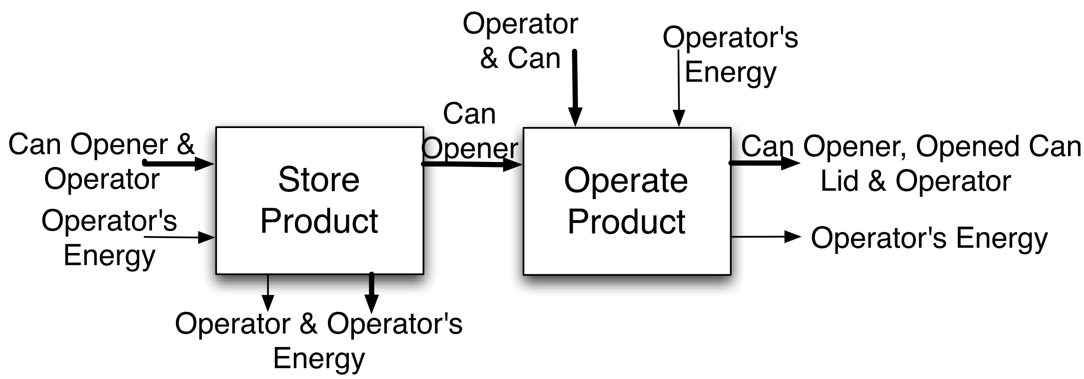


Figure 7.4 Event model of the operation of a manual can opener.

Configuration models are now created to decompose each event. The configuration model of the event *remove can lid* shown in Figure 7.5 captures the operator collecting up the can opener, *couple human & can opener*, before coupling the can opener to the can. Once coupled, the can's lid, modeled as solid material, is removed. To monitor the removal, the operator detects the progress of the lid's removal. The operator can use the status to guide actions, and once the lid is fully removed, the can and the can opener are divided. Following the operation, the can opener, opened can, lid and the operator flows are exported from the system.

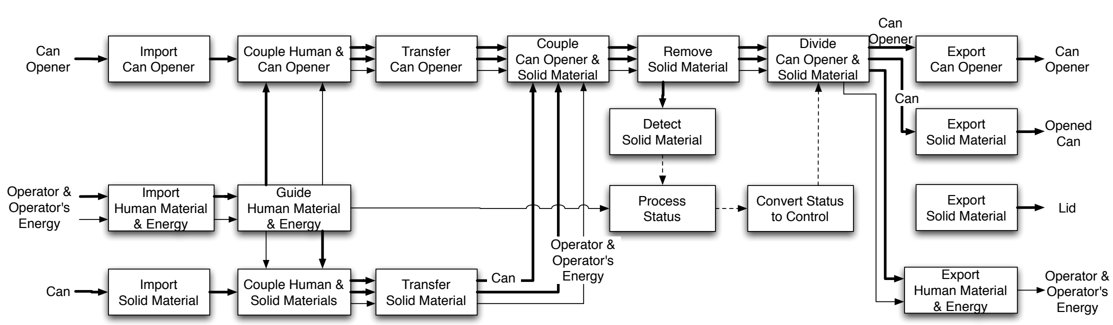


Figure 7.5 Configuration model for the remove can lid event.

In Step 3, the customer needs are translated to outcomes. These outcomes are then correlated to the process and functional models of the reverse engineered product and manual process. A potential correlation between customer needs to outcomes to function-flow pairs is provided in Table 7.1. The function-flow pairs in Column 3 of Table 7.1 are extracted from the previously generated functional and process models for the product to illustrate that the identified customer needs are met by an existing process or functional element.

Table 7.1 Correlation between needs, outcomes and functionality.

Customer Need	Outcome	Function-Flow Pairs
Can lids need to be easier to remove	Minimize the effort required to remove can lids	• Convert Human Energy to Mechanical Energy • Transfer Mechanical Energy • Remove Solid Material (lid)
Cans need to be easy to align	Minimize time required to align cans	• Guide Solid Material (can) • Position Solid Material (can)
Opened cans should separate easily	Minimize the effort required to remove open cans	• Divide Can Opener & Solid Material (can) • Export Solid Material (lid) • Export Solid Material (can)
Operation should begin when the can is placed	Minimize the time required at the start of operation	• Detect Solid Material (can) • Process Status (can) • Actuate Energy
Operation should cease once the lid is removed	Minimize the amount of overshoot once a can lid is removed	• Detect Solid Material (lid) • Process Status (lid) • Actuate Energy

Now, functional models for the conceptual system are generated. First, following Step 4, a black box functional model is produced. Since both manual can opener and the automated can opening product share the same objective—remove a lid from a can—the black box functionality remains the same between the two. The only change that must occur is with the energy flows. For the black box functional model of the automated version shown in Figure 7.6, the secondary flow, *human energy*, is abstracted to its primary type, *energy*, to represent that the energy source is not yet known.

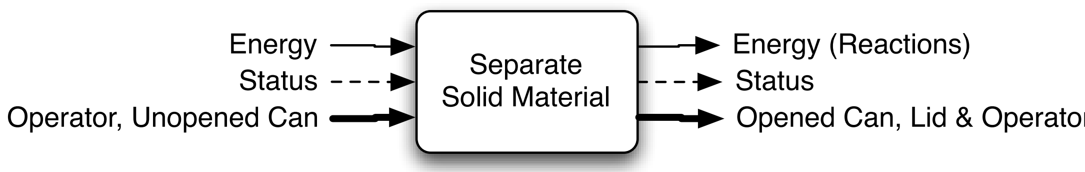


Figure 7.6 Synthesized conceptual black box model for an automated can opening product.

Step 5 directs the decomposition of the black box model into a functional model describing the specific transformations occurring to each flow. The methodology discussed in Chapter 3 is followed to generate Figure 7.7.

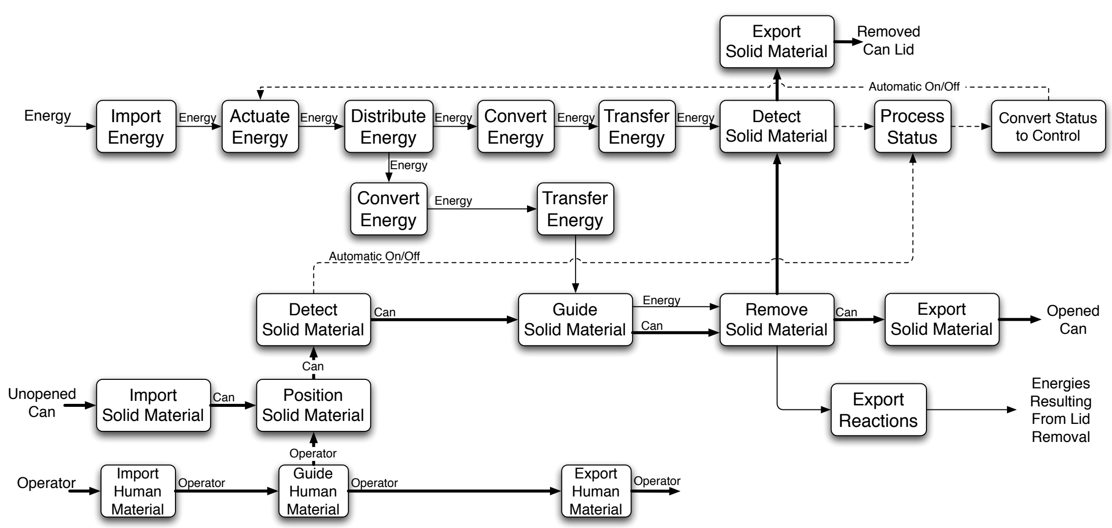


Figure 7.7 Synthesized conceptual functional model for an automated can opening product.

During the decomposition of the black box functional model in Figure 7.6 to the functional model in 7.7, specific functionality is extracted from the functional and process models generated for manual can opener. Both the functional and process models describe the importation of the can, positioning for lid removal, and exportation of the lid and open can. These mobility function-flow pairs are borrowed from the manual process to describe how the operator and can should interact in the new concept. From the process model, functionality describing the detection and processing of the status of the can is borrowed; the operator, however, no longer performs this detection. Instead, human energy is replaced with energy to specify an unknown energy source. Also, from the functional model, the transfer and conversion of energy are borrowed, and again, the flow of human energy is represented with the primary level term, energy. To complete the energy flow function chains, the function blocks, actuate energy and distribute energy, are added addressing the customer need for the can opening process to automatically start and stop. To activate the automatic on/off capability of the can opener, the can is detected upon placement into the automatic can opener (modeled as *detect solid material*) and the status of the lid removal is detected during operation (also modeled as *detect solid material*).

During the final step, Step 6, the conceptual functional model is verified to ensure that the anticipated outcomes are being addressed. Verification focuses on ensuring that the mappings identified during Step 3 are present in the functional model. If not all of the customer needs are met, the functional model should be iterated by adding new functionality or borrowing functionality from the existing product's process and functional model. For the automated can opener example, fourteen function and flow pairs were listed in Column 3 of Table 7.1 during Step 3. A simple accounting check reveals that *divide can opener & solid material* mapped to the outcome *opened cans should separate easily* is not included in the new functional model indicating that functionality is still required to separate the can, can lid and can opener product. The functional model is now updated with the function-flow pair *divide solid materials* in Figure 7.8 to represent not only the lid being cut from the can but also being removed from the system.

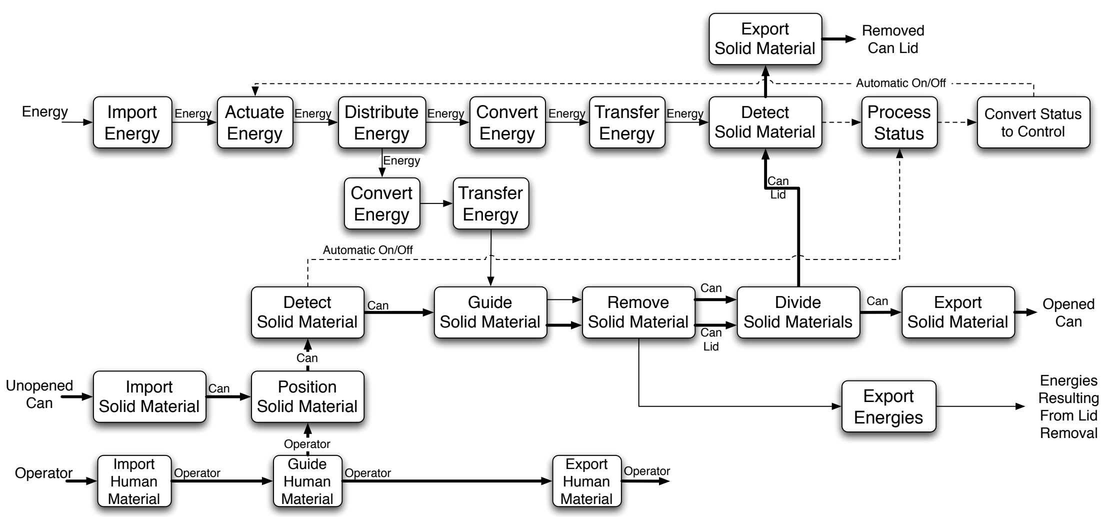


Figure 7.8 Synthesized conceptual functional model for an automated can opening product updated with the Divide Solid Materials.

7.4 Concept Generation

The functional model generated during the conceptual design of a product creates a starting point from which the design can spring board into subsequent conceptual design activities. Concept generation can be visualized through the can opener example. During conceptual design, the energy source for the can opener is unknown, thus the motive power source is broadly defined as energy. Broadly defining a flow at the primary level of the Functional Basis allows for a wide variety of alternatives to be considered during conceptual design. For instance, the can opener could be wind-up with a spring to store the energy, or solar powered with a battery to store energy, or use a laser to cut off and remove the can lid. Once a solution principle is chosen for each functional transformation, the functional model should be updated to reflect additional functionality required for the chosen solution principles.

Three common methods may be used in conjunction with functional models for concept generation including morphological analysis (manual or automated), Configuration Flow Graphs (Kurtoglu and Campbell 2009) and MEMIC (Morphological Evaluation Machine and Interaction Conceptualizer) (Bryant, McAdams et al. 2005; Bryant, Stone et al. 2005). With morphological analysis solution principles are drawn from either intuitive sources or directed search (e.g., a Design Repository (Bohm et al. 2008; Design Engineering Lab 2008)) and paired to the function-flow pairs as illustrated in Table 7.2 (Zwicky 1969; Pahl et al. 2007). Identified solution principles may be mixed-and-matched to arrive at multiple unique solutions for a design problem.

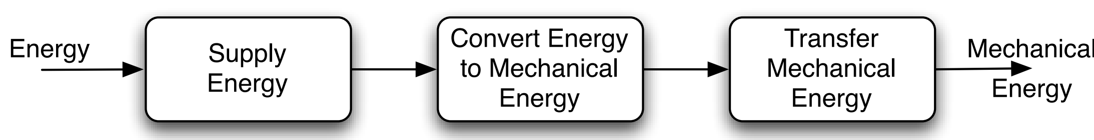
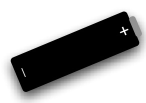
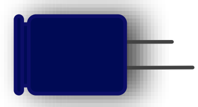
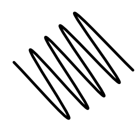
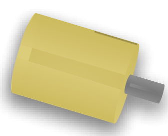
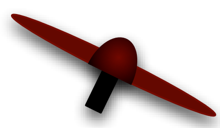
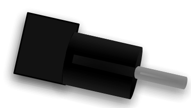
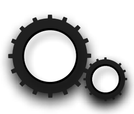
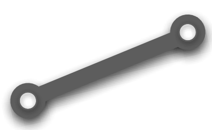
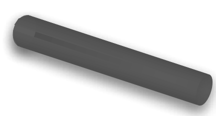


Figure 7.9 Function chain example for morphological matrix.

For example, consider the morphological matrix in Table 7.2 based on the function chain in Figure 7.9. Energy is first supplied to the system. The energy is converted to mechanical energy and then transferred from the system. The three functions represented in Figure 7.9 are listed in the first column of the morphological matrix. Solution principles are then paired to each function. Table 7.2 shows three potential solution principles for each of the functions. For instance, solution principles identified to supply energy include a battery, a capacitor or a spring. To convert the energy to mechanical energy, a motor, a propellor or a solenoid are identified, and to transfer the mechanical energy, solution principles identified include a gear set, a linkage or a shaft. Solution principles are chosen for each function to generate concepts to solve the design problem.

Table 7.2 Example morphological matrix.

	Solution Principles		
Supply Energy	 Battery	 Capacitor	 Spring
Convert Energy to Mechanical Energy	 Motor	 Propellor	 Solenoid
Transfer Mechanical Energy	 Gears	 Linkage	 Shaft

In Figure 7.10, three components, a battery to supply energy, a motor to convert the energy to mechanical energy and gears to transfer the mechanical energy, are selected from the morphological matrix in Table 7.2. These selected solution principles are mapped to each of the functions to illustrate how concepts can be selected from the morphological matrix and formulated into a complete concept.

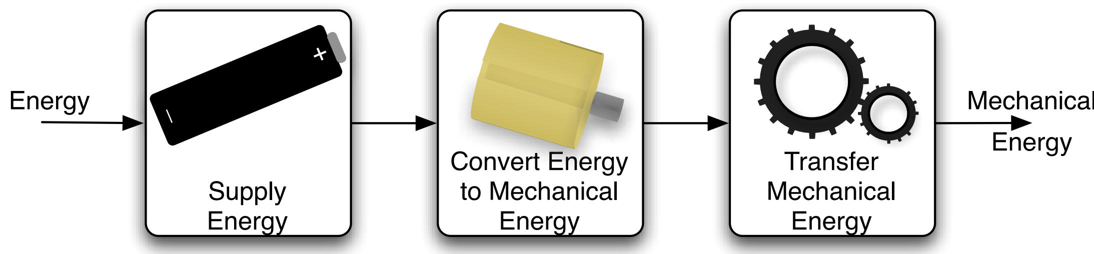


Figure 7.10 Functions in the function chain example mapped to solution principles selected from the morphological matrix.

MEMIC and configuration flow graphs automate this concept generation process. Using MEMIC requires the user to input a functional model in matrix form describing adjacency between function-flow pairs. The input undergoes a series of matrix multiplications mapping solution to functionality and filtering out component-to-component connections that are not possible based on Design Repository data (Bryant, McAdams et al. 2005; Bryant, Stone et al. 2005). Configuration Flow Graphs are also based on Design Repository data, but instead use grammar rules to convert traditionally drawn function structures into a graph of connected components. The graph is based on nodes and arcs to represent function and flow. The openness of the graph-based representation allows for automated concept generation unconstrained by singular function-to-component mappings (Kurtoglu and Campbell 2009). Any combination of these concept generation approaches is valid.

Consider now the functional model for the automatic can opener example as a starting point for concept generation. Using the automated morphological matrix from the Design Repository, solution principles from prior systems may be identified and paired to the functionality to develop a concept. In Figure 7.11, functionality has been removed from the function blocks of the automatic can opener functional model and replaced with potential solutions from the Design Repository. The resultant concept uses a plug and cord to bring electrical energy into the system. A switch activates the system. A circuit board distributes the electrical energy which is converted to a suitable voltage via resistors and transferred through the system with wires. IR sensors detect the presence of the can and the proximity of the blade to full lid removal during operation. Processors control the actuation of the system based on the signals from the IR Sensors. Once lid removal is complete, a magnet separates the lid from the can. A basket holds the can lid, and the open can is exported from the system via a guide.

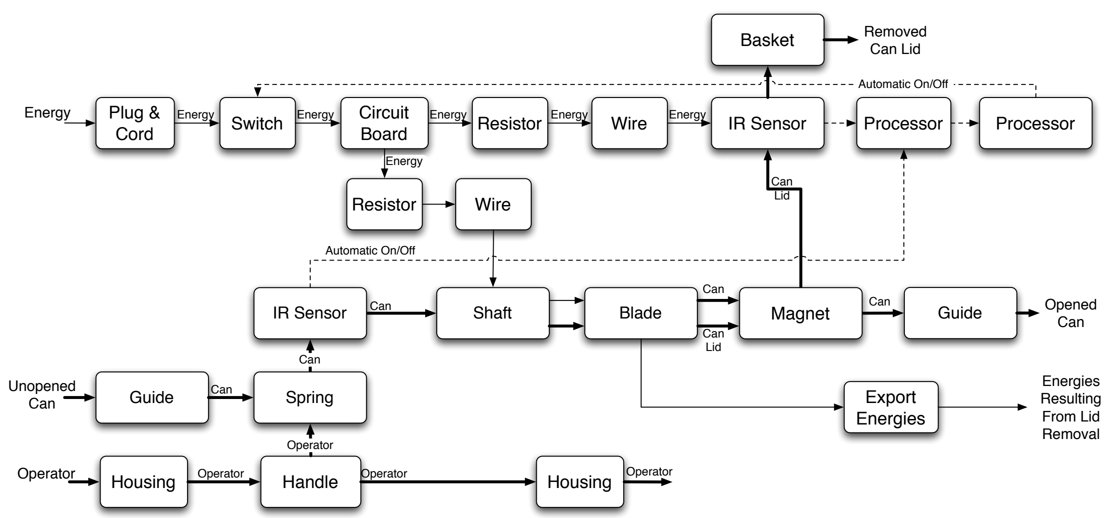


Figure 7.11 Potential component mapping resulting from a query of the Design Repository using the automated morphological matrix tool.

7.5 Comparing an Existing Model to the Synthesized

The synthesized model may now be compared to an actual product. The functional model for the conceptual can opener in Figure 7.8 can be compared to an existing automated can opener (shown in Figure 7.12) found in the Design Repository (Design Engineering Lab 2008). Both the existing and the concept have similar black box models sharing the same high-level functionality, *separate solid material*, and many of the same flows. The operator has been more specifically called out as hand and the flow of energy in the concept has been replaced with the secondary flow, electrical energy, in the existing product. The functional models also share similar functionalities; both the concept and the existing product rely on the operator to position the can for operation, and once the lid has been removed, both devices trigger an automatic actuation of energy to stop operation.

There are differences as well. The existing product is less automated than the conceptual model. The actual product instead relies on the operator's hand to secure the can into place while the concept uses the energy of the device to guide the can into the appropriate removal position. Also, there are flows in the existing product such as, change electrical energy, convert electrical energy to mechanical energy, change mechanical energy, and transfer mechanical energy, to deal with the electrical energy flow. These flows dealing with specific implementation have yet to be considered in the conceptual functional model, and would not be directly considered until concept generation such as was illustrated in Section 7.4.

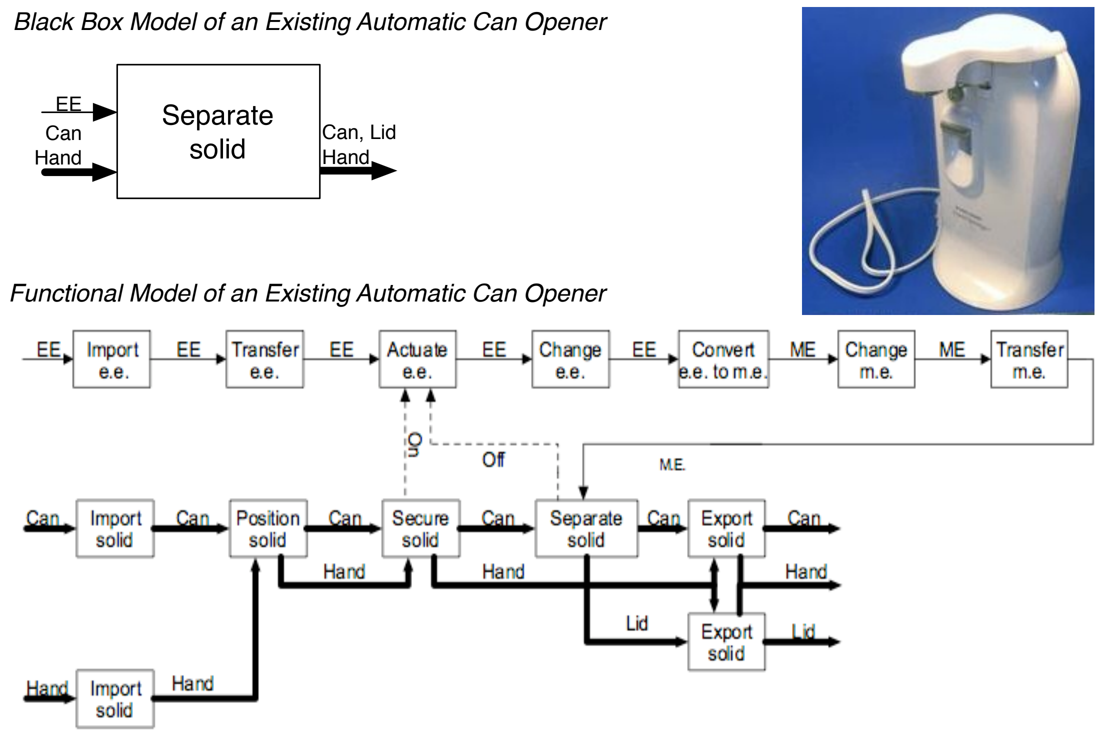


Figure 7.12 Black box and functional models for an existing automatic can opener (Design Engineering Lab 2008).

If the design process were to continue from the component to function mapping identified in Section 7.4, then as solution principles are paired to functions, auxiliary functions would also be identified, aggregated into the functional model, and new solution principles would be paired to the auxiliary functions. This would occur iteratively until a concept is fully developed. It is then possible that the updated functional model for the new automated can opener would more closely resemble the can opener found in the Design Repository. Depending on the customer needs, chosen solution principles and identified auxiliary functions, it is, however, just as likely that the can opener may share very little in common with the existing product in the Design Repository at the completion of the design process.

7.6 Summary

During the design of automated systems, it is important to fully understand the current process taken by the customer. The methodology presented in this chapter guides the use of functional and process modeling to explore existing manual processes and the products currently being employed by the customer. Human actions are captured via process modeling and products are captured via functional modeling. These models provide a starting point for the creation of a hybrid functional model that borrows from both to create a more automated solution meeting customer needs previously unmet. Core functionality of a product can be transferred to an automated system, while human actions, sensing and processing identified in the process models can be converted to functional representations. Through concept generation, mechanized systems are identified to replace the existing manual process, thus functional and process modeling provide a means to explore manual processes for automation opportunities.

CHAPTER 8 Framework for Identifying Automation Opportunities

A *framework* provides the structure necessary to methodically study an opportunity ("Framework" 2001). It is the basic structure underlying a methodology. A *methodology* is a system of methods ("Methodology" 2001). In the case for the identification of automation opportunities, the framework is the structure underlying the design methodology of automating human-centric, manual activities. Before, however, we talk in depth about this framework, it is necessary to return to the original objective and its scope. The objective, as stated in Chapter 1, is to identify automation opportunities through the investigation of potential failures related to operator error, and the scope for this research is constrained to the first two phases of the engineering design process (Figure 1.1), *Understanding Opportunities* and *Conceptualizing Solutions*. Relating this back to the context of the framework, we can say that the *methodology* structured within the *framework* will guide a designer to discover automation opportunities for the design of products that replace error prone human-centric tasks.

More generally, the framework defines the design tasks required during the first two phases of the design process to arrive at an automation opportunity and finds manual activities ripe for automation that include, but are not limited, to those that are error prone. This general framework, shown in Figure 8.1, begins with the identification of customer needs. These customer needs are mapped into the engineering domain where they can be used to drive the engineering design process. Once in the engineering domain, translated customer needs are used to systematically identify automation opportunities. These automation opportunities are then used as the inputs to the conceptualization of solutions during concept generation.

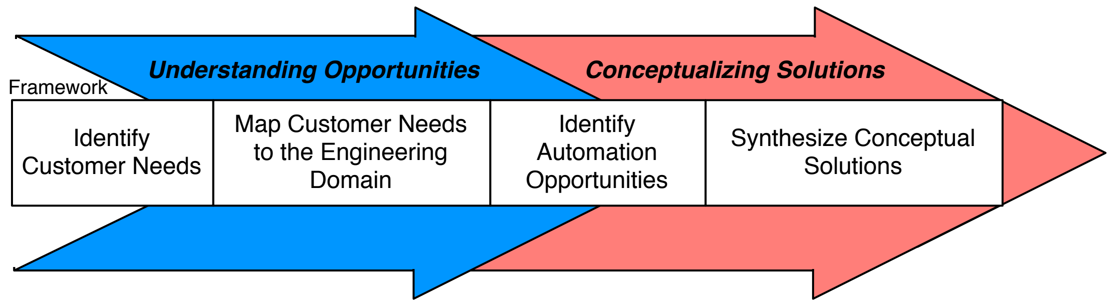


Figure 8.1 Framework for the identification of automation opportunities.

More specifically, the framework gives the underlying structure for the design methodology to automate error prone human-centric actions.

8.1 The Framework and A Methodology

Each of the tools discussed in the prior chapters may be combined into a complete methodology to address each of the four stages of the framework illustrated in Figure 8.1. Each of the tools comes together to answer the specific objective of discovering automation opportunities for the design of products that replace error prone human-centric tasks. Yet the methodology also fits within the framework defined by the underlying objective to identify automation opportunities within human-centric processes. Figure 8.2 illustrates this mapping between the *methodology* and the *framework*.

The methodology begins by directing a design team to initially understand opportunities related to the customer needs. For this, customer inputs defined by Ulwick's Outcome-driven Method are used (Ulwick 2002; Ulwick 2005) following the method demonstrated in Chapter 5. These inputs may be translated to the engineering domain using Integrated Functional and Process Modeling by following the method demonstrated in Chapter 4. Before synthesizing solutions, however, process-based failure propagation and the associated impact factors are used to rank order failure points by following the method demonstrated in Chapter 6. From these failure points, we get the automation opportunities that are used for concept generation. Two approaches are used to guide the synthesis of conceptual solutions for automating these failure prone processes; both are discussed and examples are presented. Performing concept generation for these error-prone processes allows a designer to systematically identify strategies and/or engineered solutions that can replace the human-centric tasks.

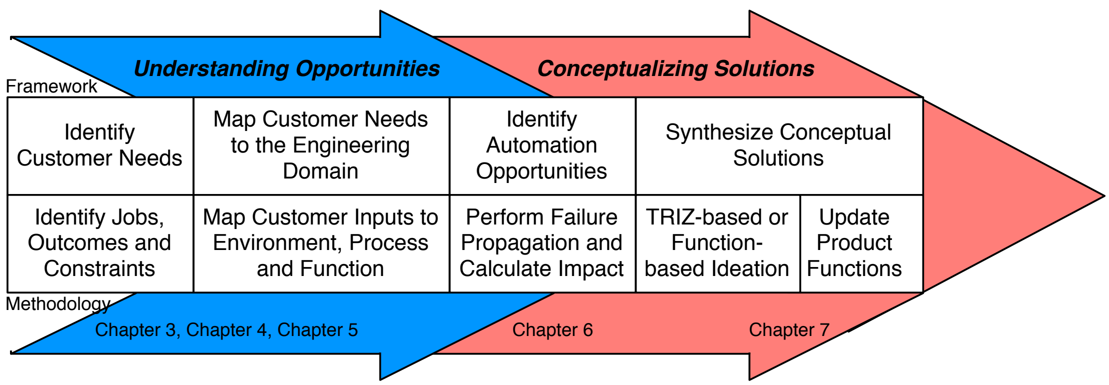


Figure 8.2 Methodology for the identification of automation opportunities.

Each of the following subsections describes one of the specific stages in the framework and details the specific methodological steps required to systematically assess a product design for automation opportunities based on error prone human-centric processes.

8.1.1 Identify Customer Needs

Framework: The first stage of the framework focuses on the identification of a proper set of customer needs. As discussed in Chapter 5, collecting an appropriate set of customer needs provides the building blocks from which a successful engineered design may be constructed. A number of techniques may be used to elicit a set of needs from the customers including focus groups, interviews, questionnaires, being the customer, and ethnographic or observational studies (Otto and Wood 2001; Ulrich and Eppinger 2004; Ullman 2010).

Methodology: To elicit a set of customer needs, the methodology uses the customer inputs defined by the Outcome-driven Method. (See Chapter 2). These specific inputs consisting of *jobs, outcomes* and *constraints,* are based on the circumstances surrounding the tasks which cause a customer to need a product and therefore define the underlying reasons as to why customers purchase products (Ulwick 2002; Ulwick 2005). Use of the following three methodological steps from Chapter 5 will guide this stage of the framework:

1. Gather customer needs using a technique such as focus groups, interviews, ethnographic studies, or surveys to create a compiled list of customer needs statements. In particular, the objective is to classify these customer need statements as either specific *jobs*, *outcomes* or *constraints* for the identified product opportunity.
2. Formulate the raw customer needs statements into specific statements of jobs, outcomes and constraints. Customer need statements are typically representative of the outcomes and constraints associated with the jobs the product is to perform. *Jobs* are formulated as functional (task to be completed) or emotional (feeling to be de-

rived) statements of purpose (Ulwick 2005). *Outcomes* are structured as phrases taking the form: *direction / unit of measure / outcome desired* (Ulwick 2005), and *constraints* are formulated as statements that summarize an obstacle preventing product adoption.

3. Sort the formulated job, outcome and constraint statements using an affinity diagram (Otto and Wood 2001). Jobs should be used as the primary categories in the affinity sort. Outcomes and constraints should be grouped with their associated job categories.

8.1.2 Map Customer Needs to the Engineering Domain

Framework: During the second stage of the framework the customer needs are transferred into the engineering domain. Mapping customer needs to the engineering domain is key to developing a successful product, as customer needs drive an innovative design process. For example, in Quality Function Deployment, customer needs (termed attributes) map to engineering characteristics with the House of Quality, and through four distinct houses, these customer needs are linked to production requirements (Hauser and Clausing 1988). In Axiomatic Design, the customer domain maps to the functional domain, the functional domain maps to the physical domain and the physical domain maps to the process domain resulting again in customer needs being linked to the final outcome of the design process (Suh 1990; Suh 2001). In Function-based Design, customer needs map to flows in a function structure, and it is the transformation of these flows that link the customer needs to components and components to designs (Pahl et al. 2007).

Methodology: Specifically, the methodology uses the integrated functional and process modeling method defined in Chapter 4, to map customer needs into the engineering design. The customer needs will be mapped into the engineering domain by answering the questions: *Who* will use the product? *Where* the product will be used? *Why* the customer needs the product? *When* and how long the customer will expect to use the product? And, *what* the customer expects the product to do? Use of the following steps from Chapter 4 will further guide this stage of the framework:

4. Identify *black box process* and *events* from the jobs. The black box process abstracts (and therefore maps to) the overall customer goal to define the product opportunity. Events abstract to the ancillary tasks required to complete the overall customer goal.
5. Identify *flows* including the product being designed from the outcomes and constraints. Map each flow to the specific events where they are required. Add new ancillary tasks when flow-event mappings cannot be found.
6. Formulate an environment model from the customer inputs to define where the customer will use the product. This model defines the location where all flows will be derived as well as all operating conditions to which the product will be subjected.
7. Formulate a black box process model from the overall customer goal identified in Step 4 and the flows identified in Step 5. This model defines the overall usage of the product being designed along with its associated energy, material and signal inputs and outputs.

8. Formulate an event model to consist of chains of events connected by flows of materials, energies, and signals that must be completed systematically to achieve the desired goal. The event sequence should begin with the initial product action and should be followed by all other discrete events including each of the actions, environments or situations where the product will be used over time. The product should be included as a flow through each event.
9. Decompose each individual event in the process model into a configuration model detailing the discrete changes to the product and any associated functional interactions with other flows in the event. Begin by generating chains of configurations. Once configuration chains capture changes from input to output, aggregate chains into complete configuration models.
10. Formulate a black box functional model to fulfill the circumstances abstracted by the process model. Functional models may be independently generated for each configuration change modeled in Step 9. Typically, however, functional models aggregate all operational aspects of a product design into a single functional model. This is a choice that is left to the design team. Flows required in the functional model should be derived from flows in the process model.
11. Decompose the black box functional model into functional models detailing the discrete changes to each flow within the boundaries of the product. Begin by generating chains of functions. Once function chains capture changes from input to output, aggregate chains into a complete functional model.

8.1.3 Identify Automation Opportunities

Framework: The third stage of the framework investigates the engineering translation of the customer needs for automation opportunities. The number of potential avenues for investigation are countless, but perhaps the most logical include investigation of the positive and negative impacts associated with automated systems. For example, discussion in Chapter 1 discussed negative aspects including physiological issues, loss of skill, and loss of accountability as well as positive aspects including increased safety, improved reliability, repeatability, and convenience, and reduced effort and down time (Sheridan 2002). Investigating an engineering translation of the customer needs for these negative and positive impacts might identify places both appropriate and inappropriate for automation.

Methodology: In particular, the methodology focuses on error prone human-centric tasks with the postulation that minimizing error prone tasks improves task reliability and safety—two of the potential benefits of automated systems. To identify these error prone tasks, propagated failure analysis and the associated qualitative and quantitative impact factors (propagated fault levels and system to process sensitivity) are applied following the steps discussed in Chapter 6. The following steps guide this stage of the framework:

12. Perform *Propagated Failure Analysis* by first eliminating failures at flow importation points, converging, and branchings in the configuration models for the product being designed. Propagate these failures along flow paths through each configuration model, and determine if interactions between eliminated flow and other function-

flow points result in child failures. For each child failure, repeat the failure propagation through the configuration model.

13. Determine the impact factor of each failure by using qualitative and/or quantitative analysis techniques.
    a. Determine the qualitative impact by assigning one of the following failure levels to each identified failure: (1) Process Terminal, (2) Process Transient, (3) Event or (4) Configuration.
    b. Calculate the quantitative impact factor as the ratio of failed configurations with respect to operational configurations over a region of interest in the configuration models of the product. Use a statistical stack-up algorithm to calculate the sensitivity profile of the system from the impact factors.
14. Rank order impact factors. For qualitative impact factors, tabulate failures by impact to the system as a whole beginning with Process Terminal Failures and ending with Configuration Failures. For quantitative impact factors, generate a Pareto Chart to rank order failures from highest to lowest sensitivity. Calculate and plot a cumulative percentage to identify the 80-20 threshold. As automation opportunities, consider first those failures with a Process Terminal impact and those falling within the 80-20 threshold.

8.1.4 Synthesize Conceptual Solutions

Framework: During the fourth and final stage of the framework, automation opportunities are explored creatively to synthesize conceptual solutions. The goal of concept generation is to generate many possible solutions.

It is through the identification of these possible solutions that a *best* solution may be found (Otto and Wood 2001). Concept generation activities can search internally with focused brainstorming and 6-3-5 or externally by talking with experts, reviewing patent documents, exploring design catalogs, trade journals and research literature (Ulrich and Eppinger 2004). Solution principles gathered by the team may be tabulated in a morphological matrix (Zwicky 1969) as illustrated in Chapter 7. This represents a form of partial solution generation where by solution principles may now be paired together to form multiple concept variants. Alternatively automated search tools based on the Design Repository (Design Engineering Lab 2008) can assist with developing complete concepts from prior product data. Tools to assist with this automated directed search include the Automated Morphological Matrix, Configuration Flow Graphs (Kurtoglu and Campbell 2009) and MEMIC (Morphological Evaluation Machine and Interaction Conceptualizer) (Bryant, McAdams et al. 2005; Bryant, Stone et al. 2005). Concept generation may focus on a single approach or may use multiple approaches to synthesize concept variants.

Methodology: The methodology considers two approaches for concept synthesis. Approach 1 considers the failed configurations as product functionality. Empirical guidelines (Chapter 7) may be followed to identify *hooks* in the functional model where configuration chains aggregate with functionality. When these *hooks* cannot readily be identified, Approach 2 is followed. The second approach considers each failure mode as a technical contradiction following TRIZ (See Chapter 2). Step 15 guides this final concept synthesis stage of the framework:

15. Follow Approach 1 to identify *hooks* in the functional model of the system that may be used as aggregation points for failed configuration chains. If *hooks* can be found, aggregate the configuration chain with the functional model and perform function-based concept generation. Alternatively, if *hooks* cannot be found, follow Approach 2 by considering each failure mode as a technical contradiction. The contradiction matrix should be used to associate known solution principles with the identified technical contradiction. Apply the solution principle that provides an automated system. More specifically, Approach 1 and 2 are applied as follows.

Approach 1: An aggregated functional model is created from both process and functional elements as discussed in Chapter 7. This aggregated functional model may then be used during concept generation to realize an automated design. Three more specific empirical guidelines derived from the general empirical guidelines in Chapter 7 assist with aggregation of configuration and functional elements. (1) The functional model of the current tool often has *hooks* that may be used as aggregation points for the configuration chains. Hooks are places where energy or material flows in the functional model are directly affiliated with the goal of the configuration chain. The product flow in the configuration chain may be converted to the affiliated energy or material flows. Tying configuration chains into these functional hooks can simplify the aggregation of the two models, and should be the first point of analysis when deciding whether or not to follow Approach 1 or 2. (2) Customer actions may indicate sensing, operability and mobility requirements for an automated solution. However, since functionality derived from the customer's current process is created in a different context than functionality related to product operation, the context may need to be changed to indicate that the product is *configuring* without direct operator intervention. (3) Manual processes are powered and controlled by the human operator. When converting from manual to automated, the human operator must be converted to an engineered power and control source, thus human energy—a secondary Functional Basis term—ought to be converted to its primary level term, energy, to represent an unknown energy source. Human material may similarly require conversion to its primary level term, material, to represent an unknown material stimulus in the new, automated system design.

Once a configuration model is aggregated with the functional model of the system, function-based concept generation should be followed pairing solution principles with the newly identified product functionality. A morphological matrix, discussed in Chapter 7, may be used for this purpose.

Approach 2: TRIZ (Altshuller 1995; Altshuller 2005), discussed in Chapter 2, is used directly with the process-based failure modes to identify solution approaches for each failure. To perform TRIZ, each failure must first be considered as a Technical Contradiction where one characteristic of the technical system is at odds with another technical characteristic of the system (i.e, the improvement of one technical characteristic results in another technical characteristic worsening). The two characteristics placed at odds represent an identified failure mode. Using the TRIZ Contradiction Matrix, principles known to solve the technical contradiction may be identified from the 40 total principles. Related solutions may either be used directly or may be used to inspire novel approaches to automate the process prone to failure.

Once ideas are formulated, functional elements required to realize the idea are formulated as function chains. The functional model of the system is then updated with the new function chains such that the functional model describes all necessary functional operations required for the solution. This reformulated functional model may then be used during further concept generation activities.

8.2 Example: Revisiting the Bridge Kit Student Design Project

Reconsider the bridge kit student design project first introduced in Section 3 of Chapter 4. Recall that the bridge kit has three high-level customer needs: (1) *The bridge must be constructed at a remote location from a self-contained kit.* (2) *The bridge must be transported from the construction site to a ditch where it will be positioned for crossing.* (3) *The bridge must support a team member to allow for ditch crossing.* Also recall that the integrated process and functional models generated for the bridge kit modeled the first two needs as processes assuming that the word *constructed* implicitly meant that the bridge must be manually assembled from the kit. The final need was modeled functionally. Let us now consider the configuration model for the construct bridge event as an example for the identification of automation opportunities.

For the configuration model, shown in Figure 4.7, three potential failures are identified. First, the bridge kit itself is incomplete as a flow. This failure would propagate through the entire system causing the bridge to not be constructed, positioned or used for crossing the ditch. The components missing from the kit would determine the failure level; most likely the failure level would be either a *process terminal* if the piece is custom to the bridge or *process* *transient* if the piece can be replaced readily. Second, the flow related to the human operators of the bridge could fail, thus the operator might not be available or might not have the energy required to perform the separation/ conversion tasks required. Third, the separate configuration fails because the operator is unable understand how to separate and subsequently convert the materials into the bridge. Both the second and third failures could again result in either a *process terminal* or *transient* failure level.

Now consider these three failures as automation opportunities where functionality added to the bridge could help to prevent these failures from occurring. All three failures can be related back to two characteristics of technical systems. As the *complexity of the device* improves the *reliability* of the device worsens. The contradiction matrix reveals three principles known to solve this TC: *Do it in reverse, Transformation properties,* and *Segmentation* (Altshuller 2005). The principle, *transformation properties,* suggests geometric changes as a means to guide or provide flexibility in one direction only. The system will consequently be rigid in other directions. Applying this principle to the bridge kit inspires self-aligning construction that allows the kit to quickly be separated and converted into a bridge from its collapsed form. The bridge is still manually constructed, but alignment, matching and placing of materials is now automated.

New functional elements must now be considered in the configuration model of the bridge. As the operator separates the bridge, the bridge should *guide* and *indicate* proper alignment. The updated configuration model in Figure 8.3 includes these functions as well as a status signal (text bolded and italicized) to inform the operator of proper alignment of the bridge components.

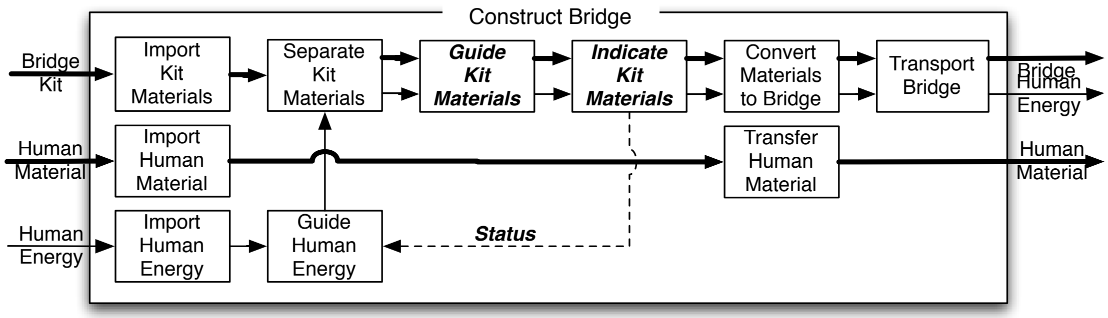


Figure 8.3 Configuration model for the construct bridge event with updates indicated with bolded and italicized text.

8.3 Example: Revisiting the Bosch CS20 Circular Saw

As a second example, reconsider the Bosch CS20 Circular Saw first introduced in Section 3 of Chapter 5. Recall that the Bosch CS20 is a professional-grade, circular saw (Bosch) that was designated one of the most innovative products of 2004 by Popular Science (December 2004).

If the configuration model of the operate product event shown originally in Figure 5.5 is considered for automation opportunities, then we again look at merging, branching and importation of flows for possible failures. An investigation of these points reveals five possible failures: (1) Material to be cut is inappropriate for the product and cannot be separated. (2) The cut path is not visible and the product cannot be guided. (3) Electrical energy flow is disrupted and the saw looses power. (4) The operator cannot provide the necessary mechanical energy to guide the product. (5) Dust and debris are not guided away from the product and therefore block the path of the saw. Each of these potential failures may be considered as design opportunities; however, let us consider just failures two and five. These failures are closely related and are identified as important to the customers who have an outcome specifying that the amount of time the cut path is blocked should be minimized (Ulwick 2005).

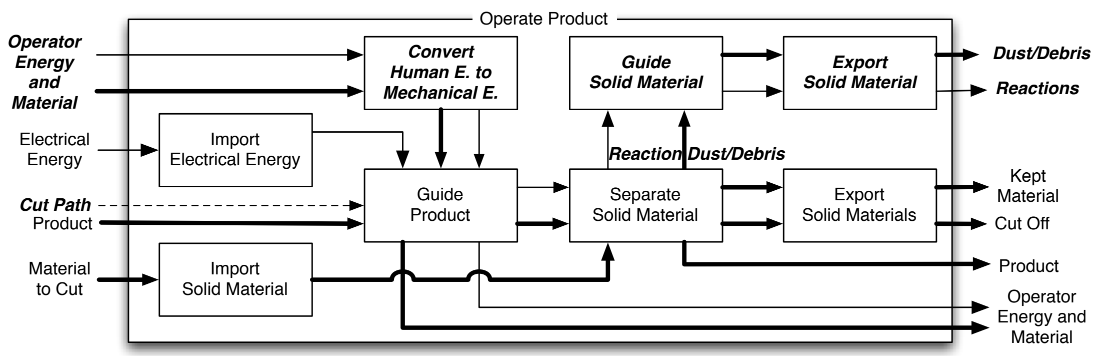


Figure 8.4 Configuration model with bolded and italicized text to indicate association with the identified failures.

To automate this configuration, approach one is taken. The configuration chains related to the failure are now extracted from the configuration model. Pieces of the configuration model in Figure 5.5 related to the failures are indicated in Figure 8.4 with bolded and italicized text. The block for the conversion of human energy to mechanical energy is extracted as it is human energy that allows the product to perform the separation. The cut path is what is being blocked, and the dust/debris is what needs to be guided and exported from the system along with reactionary (acoustic, thermal, pneumatic) energy.

These configurations considered as functions can be assembled as a single function chain as illustrated in Figure 8.5. The human energy is changed to its primary level flow energy. To illustrate that the energy type necessary to guide the solid material off the cut path is unknown, the mechanical energy is converted to energy'.

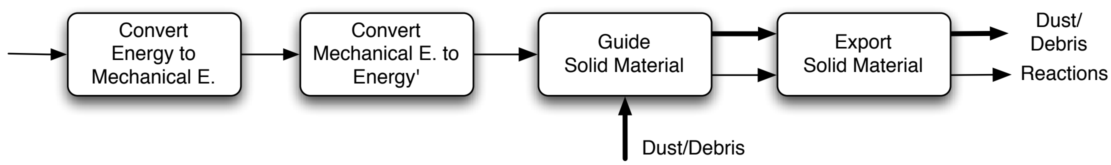


Figure 8.5 Function chain for the removal of dust and debris from the cut path.

These new functions are then used in a systematic concept generation process in the same way as all other sub-functions in the functional model. Concept variants are generated following one or multiple existing manual or automated techniques such as described in (Otto and Wood 2001; Bryant, McAdams et al. 2005; Bryant, Stone et al. 2005; Bryant et al. 2006; Bryant et al. 2007; Pahl et al. 2007; Kurtoglu and Campbell 2009; Ullman 2010). Using the automated morphological matrix (Bryant et al. 2007; Bohm et al. 2008) approach available in the Design Repository (Bohm et al. 2008; Design Engineering Lab 2008), the functions are paired with components. Table 8.1 provides an example morphological matrix using components identified from a query to the Design Repository.

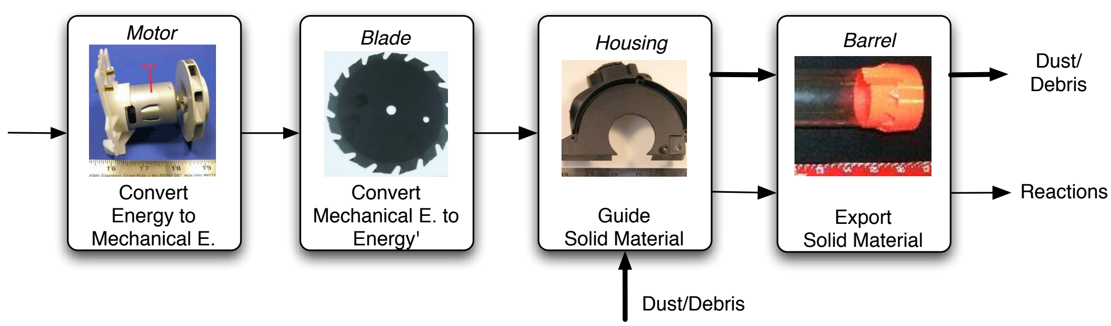
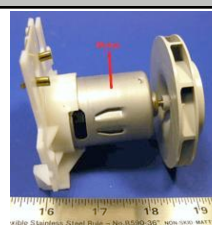
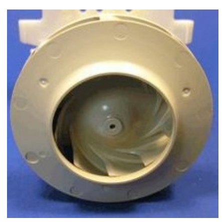
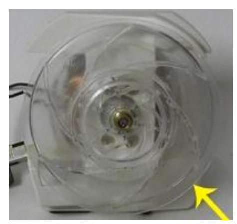
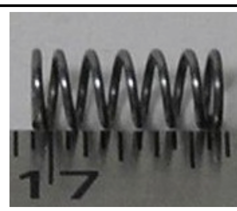
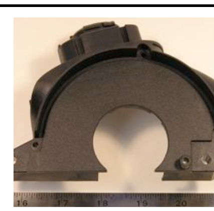
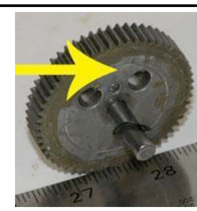
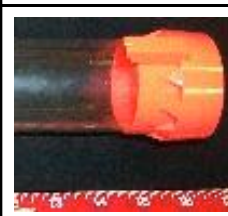


Figure 8.6 Component chain for the removal of dust and debris from the cut path.

Components are then selected from the morphological matrix to create concepts for how to rectify the failure. Figure 8.6 compiles components from the morphological matrix into the function chain (Figure 8.5) to illustrate how the configurations might be automated. This example chain uses the motor, saw blade, housing and barrel components to fulfill the new functionality required of the product. Consequently, the unknown energy source, *energy*, becomes electrical energy, while *energy′* becomes pneumatic energy.

Table 8.1 Morphological matrix for the Bosch CS20 example.

	Solution Principles		
Convert Energy to Mechanical Energy	 Motor	 Wheel	 Engine
Convert Mechanical Energy to Energy′	 Impeller	 Fan	 Blade
Guide Solid Material	 Spring	 Housing	 Gear
Export Solid Material	 Tube	 Filter	 Barrel

From the component chain, the final design might use the existing pneumatic energy generated by the motor and saw blade that is already available within the housing of the circular saw. This pneumatic energy, instead of being thrown away, would now be ducted to blow the dust and debris from the cut path keeping it clear and visible for the operator. And, in fact, this is exactly how the engineers at Bosch innovated with the CS20 professional saw. Existing pneumatic energy is ducted to remove dust from the cut path keeping it clear and visible.

8.4 On Subsequent Design Steps

Once concept generation has been performed on the aggregated functional model, the design team should have a large number of potential solution variants representing their options for solving the originally posed design problem. The next step for the design team is to sort the solution variants into feasible and unfeasible alternatives. For this a Pugh Chart (Pugh 1991) or a Decision Matrix (Otto and Wood 2001) is often used. Of the feasible solutions, those that best meet the originally identified customer needs are selected for further evaluation. Back-of-the-envelope calculations through this process will assist with validating design feasibility. As the design team hones in on the feasible design variant best addressing the customer needs, the design process moves into the embodiment phase. Philosophically, the exact boundary between different design phases are often vague. For example, the deliverable from the conceptual design phase to the embodiment phase is a final chosen solution; the design team, however, will often choose to pursue a couple design variants into the embodiment phase.

Moving into the embodiment phase of design moves beyond the scope of this text. During these last two phases of the engineering design process, embodiment and detail, we can go back to the methods discussed during Chapter 2. Tools from each of these methods along with more traditional engineering analysis should be used during these last two phases to assist with the embodiment and detailing of the final design.

8.5 Summary

A framework defining the underlying structure for a family of methodologies to automate existing manual products and processes is developed. This framework, consisting of four stages, fits within the first two phases of the engineering design process guiding the understanding of customer needs, the mapping of customer needs to the engineering domain, identifying automation opportunities and developing concepts to solve those opportunities. One methodology is presented for identifying automation opportunities based on error prone human-centric tasks. To demonstrate this methodology, two case study examples are presented.

CHAPTER 9 Case Study: Intelligent Ground Vehicle

This case study follows the design of a robot for the Intelligent Ground Vehicle Competition (IGVC) held annually at Oakland University in Rochester, Michigan (Intelligent Ground Vehicle Competition 2008). The Intelligent Ground Vehicle Competition consists of four competitions: robot design, autonomous challenge (obstacle course), navigation challenge (GPS waypoint course) and JAUS (Joint Architecture for Unmanned Systems) communication. The design of a robot for entry into the autonomous competition will be the focus of this case study.

The autonomous challenge places each robot on an obstacle course shaped like a figure eight. The boundaries of the course lane are marked with either yellow or white lines. Robots are to navigate within the boundaries of the course avoiding obstacles placed within the path. Obstacles can include construction drums and cones, five-gallon buckets, industrial trash bins, trees, et cetera. The course is typically grass; however, the robot may have to navigate around potholes, over painted wood ramps, through sand traps and over speed bumps (Intelligent Ground Vehicle Competition 2008). Figure 9.1 was taken of the course at the 2008 IGVC. In 2008, lanes were marked with white paint on the grass. A switchback required the robots to make four 90-degree turns in sequence before arriving at the intersection at the center of the figure eight. A painted, plywood ramp awaited the robots on the opposite side of the obstacle course. There were no potholes, sand traps, trees, or speed bumps with which the robots were to contend.

Figure 9.1 Navigation course from the 2008 Intelligent Ground Vehicle Competition at Oakland University in Rochester, Michigan.

9.1 Identify Customer Needs

The customer needs for the robot being designed for the IGVC are taken from the competition guide and manual. The manual describes potential obstacles, their layouts and challenges as well as the competition rules and regulations limiting the construction of each robot. This information is taken as the customer needs required to construct a successful robot.

These initial customer needs taken from the IGVC manual are considered raw needs. The first step to clarify the needs is to talk with people who have experience with the competition. For us, conversations were held with the robotics team at Missouri University of Science and Technology as they had competed in a number of prior years competitions and were familiar with the rules, challenge and judging practices. This final, clarified set of raw customer needs is compiled as Column 1 of Table 9.1.

Table 9.1 Raw customer needs, jobs and outcomes for the IGVC robot. Customer needs adapted from (Intelligent Ground Vehicle Competition 2008).

Raw Customer Need	Job	Outcome
Speed needs to remain under 5 mph during operation of the robot.	Navigate obstacle course	Maximize robot speed while remaining under the 5 mph speed limit.
All power required for the operation of the robot needs to be generated and/or stored onboard the robot.	Prepare robot to compete	Minimize the recharge/refuel time for the power system of the robot.
Robot needs to start with a single push of a button.	Prepare robot to compete	Minimize the preparation time for the robot.
	Navigate obstacle course	Minimize the effort to start the robot to a single button push.
Emergency stop needs to be readily accessible on the back of the robot.	Navigate obstacle course	Minimize the effort required to stop the robot in an emergency.
Emergency stop needs to have an operating range of at least 50 feet.	Navigate obstacle course	Maximize the range that the emergency stop can be activated to at least 50 feet.
Activation of the emergency stop needs to instantly halt the vehicle.	Navigate obstacle course	Minimize the stopping distance of the robot after the emergency stop is activated.
Robot needs to visually discern the course boundaries and standing obstacles within those course boundaries.	Navigate obstacle course	Maximize field of view on the horizontal plane.
		Maximize field of view on the vertical plane.
	Calibrate robot systems	Maximize field of view on the horizontal plane.
		Maximize field of view on the vertical plane.

Table 9.1 Raw customer needs, jobs and outcomes for the IGVC robot. Customer needs adapted from (Intelligent Ground Vehicle Competition 2008) (Continued).

Raw Customer Need	Job	Outcome
Robot needs to stay within the boundaries of the course during its operation.	Navigate obstacle course	Minimize boundary excursions during competition.
		Minimize the error when visualizing the white lines of the obstacle course.
Navigation of the entire course needs to occur in under the 5 minute time period.	Navigate obstacle course	Maximize the distance traveled in the time limit.
Robot needs to accept and carry an 18 inch by 18 inch by 8 inch 20 pound payload.	Prepare robot to compete	Minimize the effort required to load and secure the payload.
	Navigate obstacle course	Minimize the chance of loss of the payload during operation.
Robot needs to maneuver through the switchbacks placed within the boundaries of the obstacle course.	Navigate obstacle course	Minimize the turning radius of the robot.
The robot needs to operate on grass, sand and pavement.	Navigate obstacle course	Maximize the surface types on which the robot can operate.
Robot needs to avoid hitting the vertical obstacles placed on the course. These include potholes, trees, gallon pales and construction drums.	Navigate obstacle course	Minimize impacts between vertical obstacles and the robot.
		Maximize field of view on the vertical plane.
	Calibrate robot systems	Minimize impacts between vertical obstacles and the robot.
		Maximize field of view on the vertical plane.

Table 9.1 Raw customer needs, jobs and outcomes for the IGVC robot. Customer needs adapted from (Intelligent Ground Vehicle Competition 2008) (Continued).

Raw Customer Need	Job	Outcome
Robot needs to be fully or partially waterproof.	Navigate obstacle course	Minimize the entry of water into the robot.

Jobs and outcomes required for the robot are translated from each of the raw customer needs. Generally, each raw customer need represents one job and one outcome; however, in some cases multiple jobs and outcomes can be extracted from a customer need. An example of this multiplicity can be seen with the raw customer need, *Robot needs to start with a single push of a button.* Two jobs have been identified, and for each, a single outcome has been identified. Column 2 of Table 9.1 contains the list of jobs, and Column 3 contains the outcomes. A total of three jobs are identified from the raw customer needs: Prepare the robot to compete, calibrate robot sensory systems and navigate the obstacle course. These three events, compiled in Table 9.2, each represent an event required during the process of competing with the robot. The related events translate the jobs to verb-noun form.

Table 9.2 Job to event mapping for the IGVC robot.

Job	Event
Prepare the robot to compete	Prepare robot
Calibrate robot sensory systems	Calibrate robot
Navigate the obstacle course	Operate robot

The flows required for each event are derived from the outcomes by following the empirical guidelines provided in Section 3 of Chapter 5. The event-to-outcome mapping used in conjunction with the flow-to-outcome mapping provides a linking between flows and events. This linking can then be used to create a complete process model for the robot. Table 9.3 maps the outcomes and events formulated in Table 9.1 and 9.2 to the related flows required for the intelligent ground vehicle.

Table 9.3 Outcome to flow mapping for the IGVC robot.

Outcome	Event	Flow
Maximize robot speed while remaining under the 5 mph speed limit.	Operate Robot	Rotational energy, product
Minimize the recharge/refuel time for the power system of the robot.	Prepare Robot	Energy (*if battery* solid material and electrical energy), product
Minimize the preparation time for the robot.	Prepare Robot	Human material, human energy, product, status signal
Minimize the effort to start the robot to a single button push.	Operate Robot	Product, control signal, human material, human energy
Minimize the effort required to stop the robot in an emergency.	Operate Robot	Human material, human energy, product, control signal
Maximize the range of that the emergency stop can be activated to at least 50 feet.	Operate Robot	Product, control signal, human material, human energy
Minimize the stopping distance of the robot after the emergency stop is activated.	Operate Robot	Product, control signal, human material, human energy
Maximize field of view on the horizontal plane.	Operate Robot	Solid material (horizontal obstacle)
Maximize field of view on the vertical plane.	Operate Robot	Solid material (vertical obstacle)

Table 9.3 Outcome to flow mapping for the IGVC robot (Continued).

Outcome	Event	Flow
Maximize field of view on the horizontal plane.	Calibrate Robot	Status signal (horizontal obstacle)
Maximize field of view on the vertical plane.	Calibrate Robot	Status signal (vertical obstacle)
Minimize boundary excursions during competition.	Operate Robot	Solid material (horizontal obstacle)
Minimize the error when visualizing the white lines of the obstacle course.	Operate Robot	Solid material (horizontal obstacle)
Maximize the distance traveled in the time limit.	Operate Robot	Time, product
Minimize the effort required to load and secure the payload.	Prepare Robot	Solid material (payload), human material, human energy
Minimize the chance of loss of the payload during operation.	Operate Robot	Solid material (payload)
Minimize the turning radius of the robot.	Operate Robot	Rotational energy, product
Maximize the surface types on which the robot can operate.	Operate Robot	Solid material (surface), product
Minimize impacts between vertical obstacles and the robot.	Operate Robot	Solid material (vertical obstacle), product
Minimize impacts between vertical obstacles and the robot.	Calibrate Robot	Status signal (vertical obstacle), product
Minimize the entry of water into the robot.	Operate Robot	Liquid material, product

9.2 Map Customer Inputs to the Engineering Domain

Following the integrated modeling methodology discussed in Chapter 4, the environment, process and function models are generated for the robot. First, following phase one, the environments where the robot will operate are detailed. The manual for the IGVC describes how there are two areas, a main competition course and a staging area for preparation, practice and calibration activities. The overall competition occurs at Oakland University. These environments, illustrated by the double-lined boxes in Figure 9.2, are drawn hierarchically. Two sub-environments, (1) a staging area where the robot can be calibrated and prepared and (2) a navigation course where the robot actually competes in the navigation challenge, are drawn within the high-level environment, Oakland University.

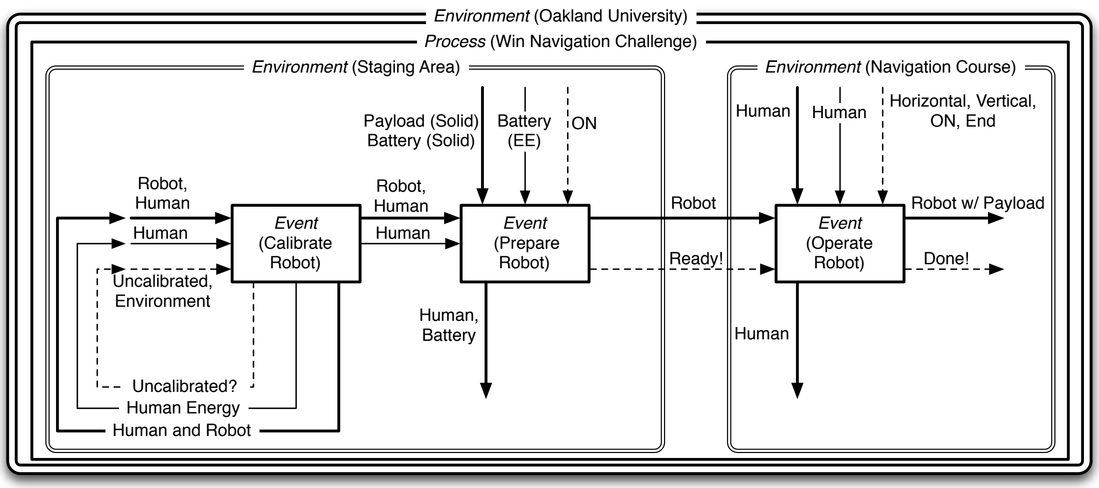


Figure 9.2 Environment, black box process and event models for the IGVC robot.

Following phase two, the black box for the process, its events and their configuration decompositions are considered for the robot. The black box event, *Win Navigation Challenge,* is created around the two sub-environments. This configuration for the models is chosen to represent that calibration and preparation within the staging area are as vital to the success of the robot as the operation of the robot on the obstacle course. Figure 9.2 places these three events within their respective environments. Flows for each event are mapped from the outcomes. To calibrate the robot, the operator flows and environment information are required as well as an uncalibrated indication from the robot. Once calibrated, the robot may be prepared. This involves charging or replacing batteries and adding a payload to the robot. In the navigation challenge, surface information, obstacle information, border information, GPS data, and, potentially, emergency stop information is required. Once done with all of the events, the robot with its payload may leave the process.

Each of the events in the process models, shown in Figure 9.2, are further decomposed as configuration models. These models for calibrate robot, prepare robot and operate robot are Figures 9.3, 9.4 and 9.5, respectively.

The configuration model for the calibrate robot event in Figure 9.3 describes the human operator interacting with the robot to actuate a calibration routine. Information from the environment, represented as a dashed line, is collected by the robot. Once this information is collected, the information is transmitted as status back to the operator. The operator can then use the information to regulate (i.e., calibrate) the robot for its operating environment.

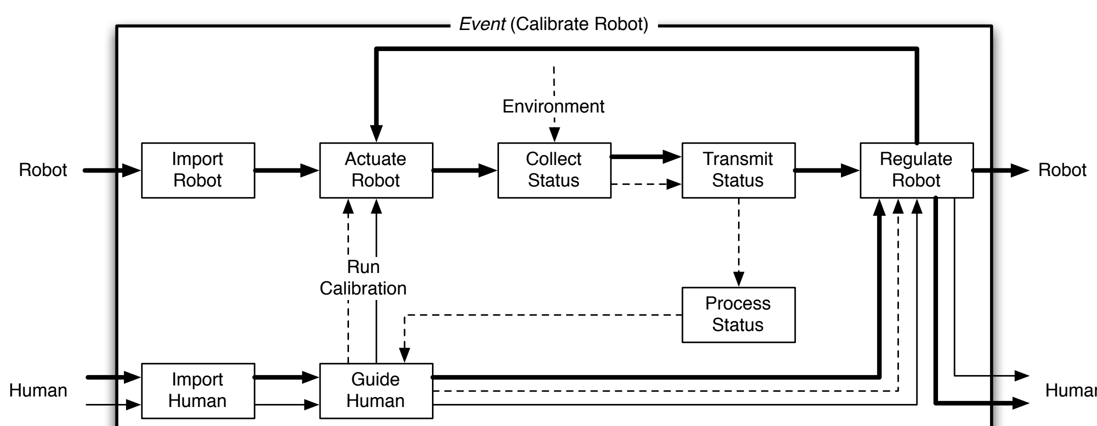


Figure 9.3 Configuration model for the calibrate robot event.

To prepare the robot for a run on the obstacle course, the operator must first change out a drained battery with a fully charged battery. The drained battery, represented by solid line, is guided from the system. Then the operator can guide a new fully charged battery into the system. This fully charged battery is represented as a bold solid material arrow and a thin energy arrow by the configuration model in Figure 9.4. The robot is then changed to accept the payload; the payload is guided into the system and coupled with the robot. This payload must remain within the system during the entire run on the obstacle course. Once preparatory actions are taken to prepare the robot, it is transported to the obstacle course, and positioned for competition. The human operator's actuation of the robot readies the software for the required single button start by the judge.

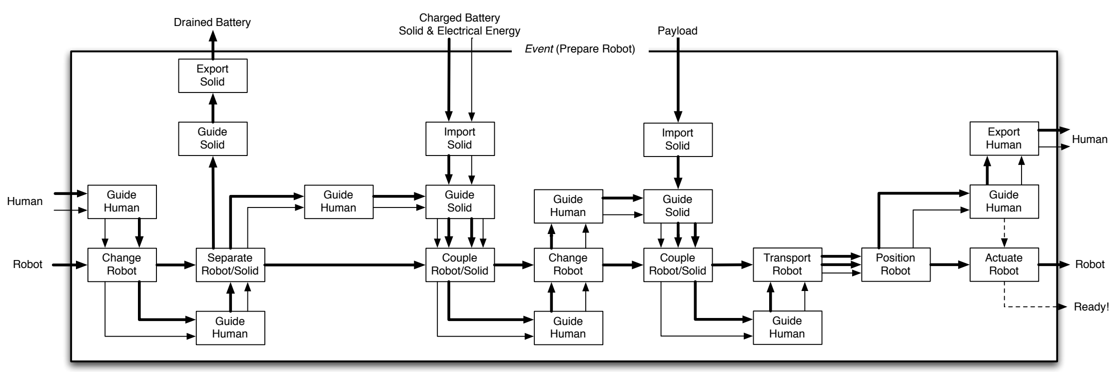


Figure 9.4 Configuration model for the prepare robot event.

In the configuration model for the operation of the robot in Figure 9.5, the judge first actuates the robot. This begins the navigation challenge. The judge waits with the E-Stop signal to either stop the robot at the end of five minutes or when an errant action occurs. The robot collects signals and processes to change itself as it navigates the course. The specifics detailing how these high level configurations occur will be detailed in the functional model as they focus on the functionality specific to the robot. A timeline represents the five minute time constraint placed on robots during this event.

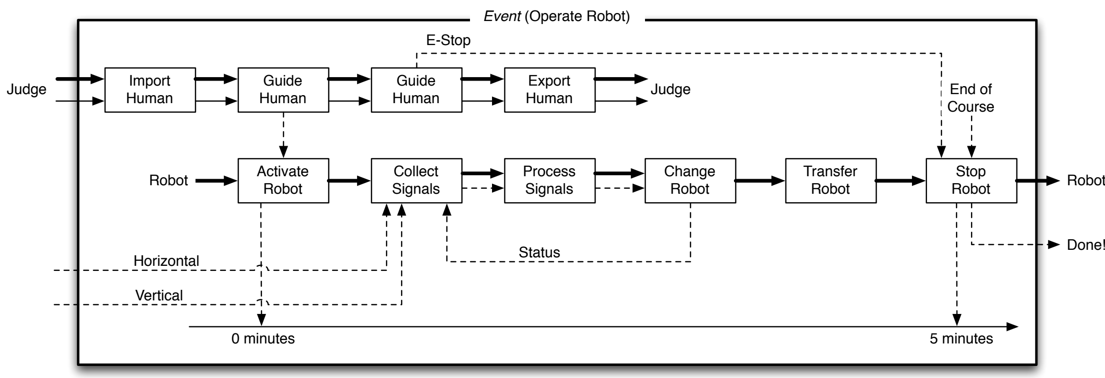


Figure 9.5 Configuration model for the operate robot event.

The configuration models are now used following phase three to guide the development of the functional model for the intelligent ground vehicle. The complete functional model included as Figure C.2 in Appendix C provides the final functional model for the IGVC robot. The functional model includes all of the flows identified in Table 9.3 and compiles the functionality required to perform the configurations modeled in Figures 9.3, 9.4 and 9.5.

The transformations for all of the identified flows required by the IGVC robot are included in the functional model. Functionality is included for the detection of solids (vertical and horizontal obstacles), processing of the status for understanding and making decisions about navigational data, actuation of electrical energy to represent the startup and termination (both intended and emergency) of the robot, motion of the robot with respect to the ground, securing of the payload, and prevention of rain ingress.

Once a judge activates the robot, electrical energy is converted to mechanical energy. The mechanical energy is regulated by the drive system controlling speed and direction of the robot. Mechanical energy is converted to rotational energy by the wheels and exported to the surface on which the robot operates. Two separate chains are included for the detection of obstacles—one for vertical obstacles and one for horizontal obstacles. These chains are processed independently first, and then collectively to determine the best path for the robot to travel. A control signal is processed to identify the best path. This path directs the direction and motion of the robot.

9.3 Identify Automation Opportunities

To identify opportunities for automation, the robot configuration models must be investigated for failure points. Points in the configuration models investigated are where flows import into the model, converge or branch. Investigation of these points reveals 17 potential, flow-based failures across all three events required for the robot to compete in the IGVC. These identified failures are separated by event in Table 9.4. The propagation start point for each failure is Column 2 of each table and the qualitative impact factor for each failure is Column 3.

Failures identified during the first event, calibrate robot, are associated with the not being able to calibrate the robot or incorrectly calibrating the robot. Failures related to calibration are isolated to the calibration event and are given qualitative impact factors of either *event level* or *configuration level.* An event level impact factor indicates that the event can not be completed, while the configuration level indicates that the event is not going to be completed at a lower level of quality than intended. Also the impact factor can change depending on the degree with which a flow is interrupted. For example, on the flow of information back to the operator, if partial data is received, the robot might be incorrectly calibrated—a configuration level failure. Alternatively, if no data is received, then the operator can not calibrate the robot at all—an event level failure. As the analysis team investigates each failure mode, these options should be considered and recorded to guide design decisions.

Failures identified in the event, prepare robot, are associated with flows required for the robot to compete. For example, not being able to add a payload into the robot disqualifies the robot for competition; this failure therefore is process terminal. Other failures include not being able to change the batteries, not being able to move the robot to the obstacle course and not being able to place the robot in a *ready* state. These failures are all given impact factors of process terminal to indicate that the robot would lose one of its opportunities to compete (i.e., a complete cycle through the event model) if the failure were to occur during the competition. These failures should be considered during the design of the robot to ensure that an opportunity to compete is not lost during the competition.

During the final event, operate robot, the failures tend toward being configuration level only. The only fault that would prevent a run of the robot, is the judge starting the robot from its ready state. Once started, the robot has begun its run; the remaining failures would not prevent the event from occurring. They would merely reduce the score (i.e., quality) of the final anticipated result. In other words, while the robot is still able to compete, it would not win the competition.

Table 9.4 Failure modes for the calibrate robot event, their propagation start points and qualitative impact factors.

Failure Mode	Start Point	Impact Factor
Calibrate Robot Event		
Interruption of the run calibration signal from reaching the robot	Converge	Event
Interruption of environment signals	Input	Configuration
Interruption of status to operator during/after calibration so that the operator does not know how the robot should be calibrated	Branch	Event (No Data) or Configuration (Partial Data)
Interruption of manual calibration signals so that the operator knows how to calibrate, but the robot will not take a calibration	Converge	Event
Interruption of the robot on subsequent re-calibrations	Branch	Event
Prepare Robot Event		
Drained batteries cannot be removed	Branch	Process Terminal
Charged batteries cannot be added to the robot	Converge	Process Terminal
Payload cannot be added to the robot	Converge	Process Terminal
Robot cannot be moved to the starting position	Converge	Process Terminal
Team cannot be place the robot in “ready” state	Branch	Process Terminal
Operate Robot Event		
Judge cannot start the robot from “ready” state	Branch	Process Terminal
Judge cannot stop the robot in an emergency	Branch	Configuration
Interruption of vertical obstacle signals	Input	Configuration
Interruption of horizontal signals	Input	Configuration
Collection of signals gets interrupted	Converge	Configuration
Changing of robot speed and direction gets interrupted	Branch	Configuration
Judge cannot stop the robot at the end of the course/time limit	Converge /Branch	Configuration

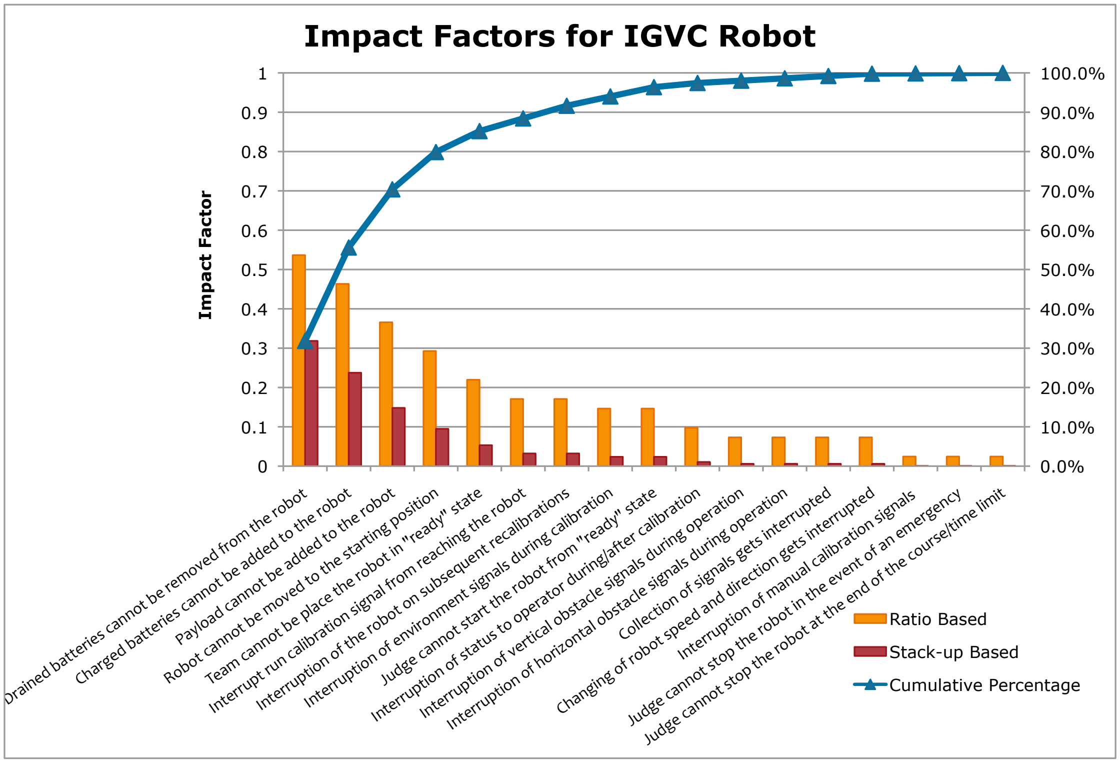


Figure 9.6 Impact factors for the IGVC robot shown as a raw ratio, scaled with a stack-up algorithm and compared as a cumulative percentage.

Once each of the failures are assessed qualitatively, a quantitative impact factor may be calculated to assess the percentage of the process affected. To identify a quantitative impact factor for each of these failure modes, a range of interest must first be selected. For the robot, since all three events are critical to success at the IGVC, the range will include all of the configurations across the three events. The three separate configuration models are aggregated into the single model. This aggregated configuration model is available as Figure C.4 in Appendix C. Failures are traced through the entire configuration model, and the percentage impact is calculated following Equation (6.1). In Figure 9.6 the raw percentage-based impact factor for each of the failures is graphed.

A statistical stack-up algorithm is used to provide insight into the relationship between each of the different identified failures in a system. Following Equations (6.2) through (6.4), each failure is compared as a percentage of the total failure profile. The sum of all failure contributions is 100% of the total failure profile. The resulting failures are plotted as a Pareto chart and combined with the raw percentage-based impact factor in Figure 9.6. In this case, four failures, representing 24% of the identified failures, account for 80% of the total variance. This again falls generally within the Pareto 80-20 rule and indicates that these failures should be considered for design improvements.

9.4 Synthesize Conceptual Solutions

Failures identified during the process-based failure analysis may now be considered during the design of the robot. Failures may be considered opportunities for automation and/or design challenges. Both of the concept generation approaches—TRIZ-based and function-based—will be applied to the IGVC robot in the following sections to address the failure modes identified from the processes.

9.4.1 TRIZ-based Concept Generation

TRIZ is used to identify potential solutions to avoid failures related to replacing drained batteries. The original idea during concept generation is to have multiple sets of rechargeable batteries that can be replaced between navigation challenge runs. Replacing the batteries during the competition, however, represents the largest potential for failure. To rectify this problem, the characteristics of the technical system that are improving and getting worse are identified from the 39 possible technical characteristics. For this failure, the decision is that the *Loss of Energy* results in worsening *Reliability*. Three solution principles are identified from the Contradiction Matrix: (1) Prior action, (2) Cushion in advance and (3) Transformation of properties (Altshuller 2005). The solution used on the IGVC robot stems from the principle, *cushion in advance.* To *cushion in advance* means that the robot should be designed to compensate (Altshuller 2005); a larger than necessary battery was used in the design of the IGVC robot. The battery, therefore, never needed to be removed because it was only partially drained. The larger battery can be charged between runs while still in the robot.

Alternatively, the principle, *prior action,* may be used to inspire an automated solution. *Prior action* tells the designer to "place objects in advance so that they can go into action immediately from the most convenient location" (Altshuller 2005). Following this principle, the design team might choose to bring the power generation system to the robot. A small portable generator, placed on the robot can cycle on and off charging the battery during the robot's operation. Not only does the battery no longer need to be removed and replaced, the team no longer needs to bring the robot to the staging area between runs to ensure a full charge for each subsequent run. And, while this solution does not completely take the operator out of the loop, it can increase the length of time between required operator maintenance points and improve the reliability of the robot.

9.4.2 Function-based Concept Generation

Second, consider failures related to the calibration of the robot. The original concept considered during process modeling involved manual calibration of the robot. The operator would place the robot in a calibration mode, and the robot would subsequently collect data on its environment. From this information, the operator could make changes to the robot's settings. Failures related to calibration would not result in an inability to compete, but would result in severely degraded performance.

The configuration chain related to calibration can be aggregated with the functional model of the robot for use during concept generation. This function chain, shown as the original Figure 9.7 (a), is changed to represent that the robot cannot be a part of its own functional model in Figure 9.7 (b). This is similar to the abstraction of human material or human energy to their primary class level in the Functional Basis when converting a configuration chain to a function chain (Discussed in Chapter 7).

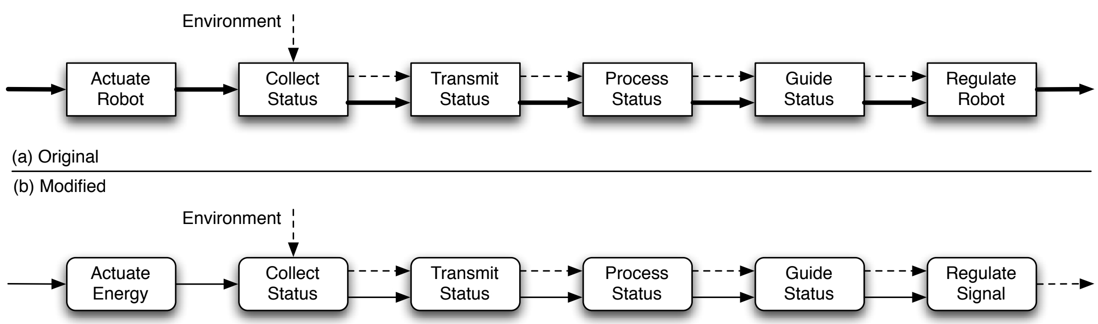


Figure 9.7 (a) Original configuration chain for the IGVC robot and (b) modified version of the configuration chain for the IGVC robot.

To modify configurations related to changes of the robot as a whole, the internal functionality of the robot must be considered. The robot, which runs on energy, is started by actuating its energy source. When converting configurations to functionality, the same can be considered. The regulation will link to how the status signals collected from vertical and horizontal obstacles are processed. To change this processing, a signal sent to the *process status* block is modified. Identifying these hooks where the configuration model will link into the functional model will provide insight into how a configuration chains should be changed to aggregate with the functional model. The new functional model containing the configuration elements are provided in Figure C.3 of Appendix C. The new functional model for the robot contains a function block for the collection of status information from the environment. This information is transmitted and processed by the robot. A status signal leaves the process block and regulates a signal that is directly linked with how the vertical and horizontal obstacles are processed by the robot. The concept is that the robot will automatically calibrate once a team member starts the robot. This will allow the robot to be calibrated correctly each time it starts a new run on the navigation course and will automate the originally manual process.

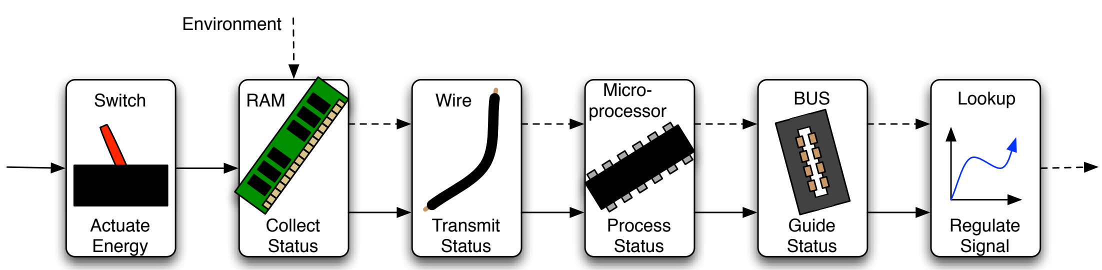


Figure 9.8 Solution principles for configurations aggregated into the functional model of the IGVC robot.

Solutions to solve each of the configurations aggregated with the functional model are generated using a morphological matrix as demonstrated in Chapter 7. In Figure 9.8 potential solution principles are mapped to each of the function blocks. A switch actuates the system, RAM is used to collect signals from the environment. A wire transmits the signal to the microprocessor where is it processed. The resulting calibration signal is guided through a data BUS that connects the calibration information to the processing units for the vertical and horizontal sensory units. A lookup table contains the proper settings for each sensory unit based on the calibration signal.

Concept generation is also performed for the entire functional model to generate possible solutions for the final design. Figure 9.9 further illustrates partial solution generation with the vision system and path determination portions of the functional model. A construction barrel represents the vertical obstacles that might be encountered on the obstacle course; these vertical obstacles are detected via a Hokuyo URG-04LX laser scanner (Hokuyo Automatic Co. 2005). White lines represent the horizontal obstacles that might be encountered and are detected by an Apple iSight digital camera (Apple Corporation 2004). The raw data from the laser scanner and the iSight are processed independently before being analyzed for potential gaps where the robot can traverse. A ray-casting algorithm is implemented for the robot's path identification algorithm.

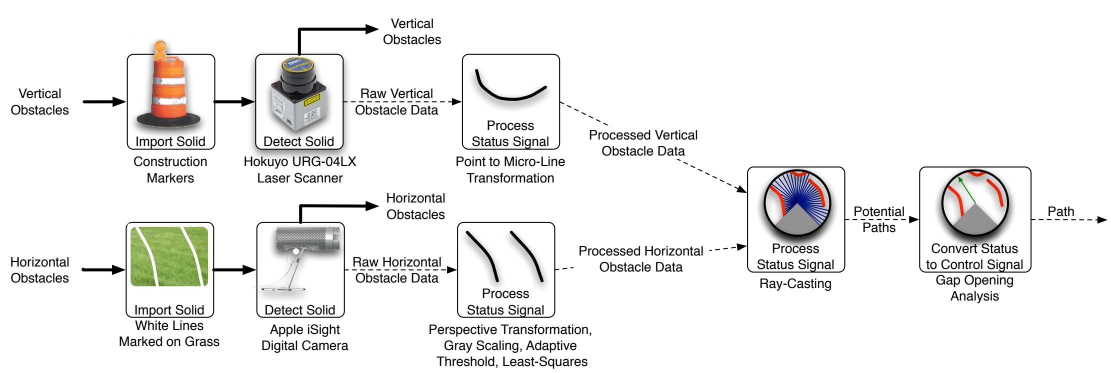


Figure 9.9 Component-to-function mapping for the vision system and path determination portions of the conceptual functional model of the IGVC robot.

Concept generation for all other portions of the functional model would be performed similarly.

9.4.3 On Addressing the Remaining Failures Modes

The remaining two failures identified through the process-based failure identification method that should be investigated include: *Payload cannot be added to the robot* and *robot cannot be moved to the starting position.* These failures, like those already solved, have the potential to prevent the robot from competing in the IGVC. The design team of the robot addressed these failures through their choice of solution principles. Instead of an elaborate payload loading mechanism, an open space slightly larger than the payload was crafted on the top of the robot. The open space was simple and very reliable. To ensure that the robot could always get to the starting position, handles fastened to the chassis of the robot allowed the robot to be pushed along the course. This prevented extraneous use of electricity and provided a very reliable, albeit simple approach to preventing the identified failure. So in a sense, these remaining failures lead to non-automated, yet effective solutions.

9.5 Summary

Following concept generation, the design team evaluated concepts first using a Pugh Chart (Pugh 1991) to weed out the infeasible concepts and then a Decision Matrix (Otto and Wood 2001) to identify the best concept. Proof-of-concepts were created to hone in on the best design variants. During the embodiment phase of the design, the team created a solid model representation for the robot to package the chosen components. The solid model, shown in Figure 9.10 (a) represents the final iteration for the design embodiment. The team then began to detail the design and develop a physical prototype for the design. During these phases the team developed the software algorithms required for the design, designed the electrical circuits necessary to power the motor system, and integrated with the electrical system on the wheelchair (the teams chosen chassis).

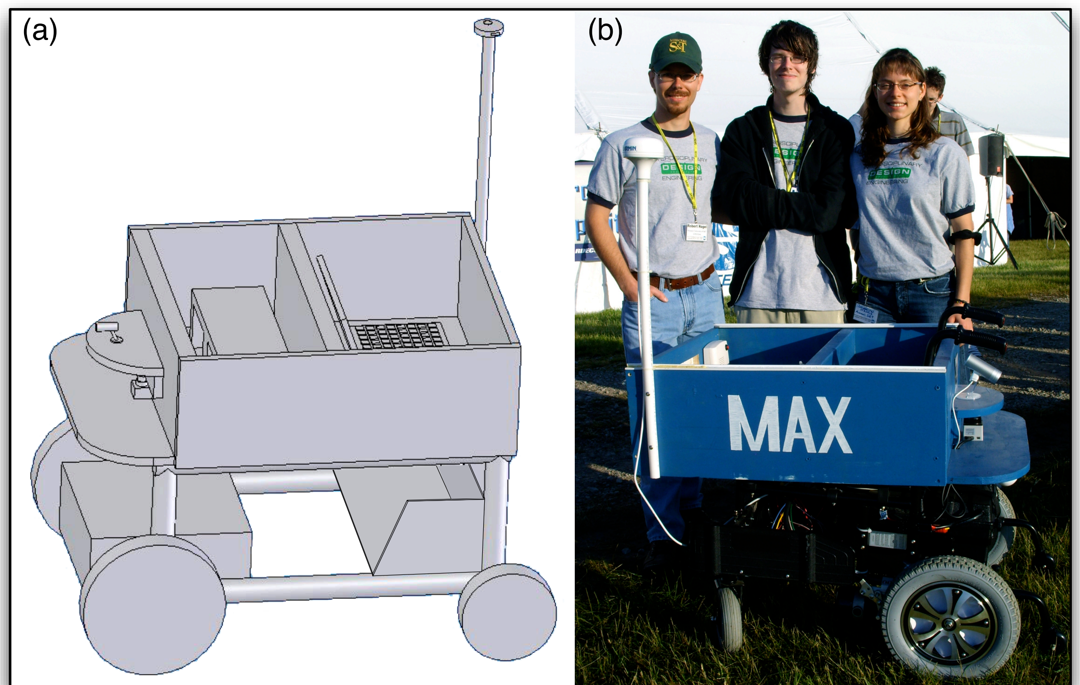


Figure 9.10 (a) Solid model of the final design of the IGVC robot and (b) the final prototype at the IGVC at Oakland University in Rochester, Michigan.

The final design, shown in Figure 9.10 (b) at the IGVC at Oakland University in Rochester, MI, competed in both the design and navigation challenge, and represented a completely different approach to the competition. Where the majority of the teams competing at the IGVC iterated on prior designs or developed complicated systems to solve the needs and requirements, our robot addressed the design from a needs-based approach evaluating the needs through the design process to ensure a design that is simple, to the specifications and not overly complicated. The final design won 6th place out of the 41 designs in the Design Challenge and was ranked 13th place out of 25 in the Navigation Challenge.

CHAPTER 10 Conclusions and Future Work

The goal of automating a system is to replace manual operations with systems, methodologies or techniques. In this way, automation can reduce human interaction or relieve human control during manual processes ("Automation" 2001). When seeking opportunities for automation, it is important to focus on the human-centric process that is to be replaced. It is these processes that can provide insight into where automation is appropriate or necessary. The investigation of these human-centric processes and the identification of automation techniques to replace these processes in the examples and case study explored in this dissertation proves the hypothesis that indicators may be identified during conceptual design to identify human-centric actions ripe for automation.

10.1 Conclusions

To systematically identify automation opportunities, a methodical design approach is formulated. This is the key contribution of this research. Through this approach, a designer can understand human-centric actions, analyze those human-centric actions, identify automation opportunities, and perform concept generation. Applying this methodology to a manual process, allows a designer to prescribe automated solutions to manual tasks starting from basic customer needs and to quantitatively predict the impact of an automated solution as it relates to an identified failure mode. Recall that six objectives are identified to develop this methodical design approach:

1. Extend function-based hierarchical models for design abstraction through integration with a process-based representation.

2. Extend outcome-driven design through integration with function-based hierarchical models.
3. Develop a formal approach to identify, propagate and rank-order failures through human-centric processes.
4. Extend function-based concept generation for the identification of solution principles of human-centric processes.
5. Develop a framework for the identification of automation opportunities in human-centric processes.
6. Demonstrate the application of the method to conceptual and reverse engineering problems through case studies.

The work of this dissertation contributes to the field of engineering design. Contributions stem from the fulfillment of the objectives listed above. Each objective is met by the following: Objective 1 is met with the integration of functional and process modeling in Chapter 4. Chapter 5 addresses the second objective by establishing links between and extending both outcome-driven design and function-based design. A process-based failure analysis approach and impact factors are developed in Chapter 6 to investigate failure propagation in human-centric processes. This failure analysis approach addresses Objective 3. Objective 4 is addressed in Chapter 7 with the methodology to pair solution principles to manual actions. The framework defining the underlying structure for a family of methodologies to automate existing manual products and processes presented in Chapter 8 addresses the fifth objective. The examples at the end of Chapter 8 as well as the detailed design case study in Chapter 9 address the sixth and final objective.

The *framework* established in Chapter 8 defines the underlying structure for a family of methodologies to automate existing manual products and processes. This framework, consisting of four stages, fits within the first two phases of the engineering design process guiding the understanding of customer needs, the mapping of customer needs to the engineering domain, identifying automation opportunities and developing concepts to solve those opportunities. A methodology supported by this framework for automating error prone human-centric tasks is presented. This *methodology* prescribes the automation of manual tasks starting from basic customer needs and quantitatively predicts the impact based on the failures replaced.

Specifically, contributions include the integration of functional and process models to abstract not only *what* a product must do, but *why* a customer needs the product. Function-based design is extended through integration with outcome-driven design to more formally map customer needs into the engineering domain. Process-based failure analysis allows for the identification and exploration of failure points in manual actions. Impact factors quantify not only the impact of a failure, but also provide insight into the resultant impact of automating error prone human-centric tasks. Ideation based on TRIZ and function-based concept generation provide the tools necessary to develop the systems required to reduce human interaction or relieve human control during manual processes.

10.2 Broader Impact

A formal approach to identification of automation opportunities in human centric tasks holds the potential to not only improve or simplify the customer's interactions with a product, but also to potentially remove the customer from harmful operations. The electric can opener example (Chapter 7) and the Bosch circular saw example (Chapter 5 and 8) illustrate how the tools presented in this dissertation can be applied to consumer products, but as illustrated with the IED example in Chapter 6, the broader impacts of this research extend beyond consumer products. With project support from both the US Chemical Corps and the Joint IED Defeat Organization (JIEDDO), process analysis is a recognized tool for the analysis of human-centric operations. With the US Chemical Corps, Nuclear, Biological and Chemical (NBC) decontamination procedures are largely human-centric with mops, buckets and equally toxic chemicals being employed by the soldiers to neutralize the NBC contaminates on the equipment (Nagel et al. 2006). Process analysis applied in this manner as described in Chapters 3 and 4 is utilized to identify automation potential in the current soldier operations; this application has the potential to save soldiers' lives.

Similarly, research with the US Air Force Academy supported by JIEDDO, uses integrated process and functional modeling, process-based fault propagation and the impact factors described in Chapter 6 to analyze IED events for fault points that can be exploited to mitigate soldier risk in hostile zones (Nagel, Greer et al. 2009). Further, research at the US Army Tank Automotive Research, Development and Engineering Center (TARDEC) and Army Research Lab (ARL) focusing on the automation of human-centric operations recognizes the importance of this type of research to the future success of our armed forces. For example, research into applications for automation are leading to a number of technologies that include: Power assist doors on heavily armored vehicles that reduce muscle strains and door-slammed fingers (Schmitz and Manceor 2009); kit-based, autonomous vehicular control devices that facilitate automated convoy mobility that allow solders to focus on surveillance rather than vehicle operation (Schoenherr 2009); automation of acoustic-based, non-destructive testing for armor that replaces the human ear to detect damaged armor sections (Williams 2009), and robotic kits tethered to construction equipment to assist troops with hastened route clearance operations (Theisen and Richardson 2009).

Beyond automation of human-centric processes are applications in complex system design. Research performed with the General Motors Research and Development Center, has led to the separation of functional chunks of complex systems, such as automobiles, via their anticipated use in the final product architecture (Nagel, Hutcheson et al. 2008). This work allows functional decompositions to be developed independently by different operational units in a system and reconnected via flows at the process layer. The whole model may be viewed or each operational unit's functional (or configurational) model may be viewed independent of the whole—a necessary feature as models increase beyond the size of a computer monitor. The development of a model at this very-high level of fidelity following the approach detailed in Chapter 4 provides the framework necessary for coarse balance activities utilizing function-based, flow-tied behavioral modeling (Hutcheson et al. 2007) where each operational group attributes reusable, mathematical models for functional attributes of final product architectures. Model inputs may then be tweaked as knobs to provide coarse balance for preliminary architecture designs, thus reducing development costs spent on unfeasible architecture balances.

Last, the tools and methods discussed herein have applications in education of engineering design. Exposure to engineering design helps students develop skills for problem solving, information synthesis and knowledge analysis and integration. An education in engineering design gives students the skills required to creatively solve real-world problems (Atman and Bursic 1996). Following the Second World War, however, these courses focusing on engineering design (as well as those focusing on shop and manufacturing methods) began to be removed from the typical engineering curriculum in favor of engineering science theory. This pendulum swing left students without the hands on design expertise required to be work-ready engineers (Dutson et al. 1997). Consequently, engineering programs were built on engineering science where analysis is the focus and mathematics is the language (Dym 1999). With direction from the Accreditation Board for Engineering and Technology (ABET) and pressure from industrial companies, engineering design has been reintroduced into the standard engineering curriculum—first through capstone (senior-level) design courses (Dutson et al. 1997), then through cornerstone (freshman-level) design courses (Dym et al. 2005), and finally, with integrations of engineering design projects during traditional, theory-based, engineering courses (Stone, Hubing et al. 2005). Design tools such as those discussed in this dissertation applied during engineering design courses (such as with the student design bridge kit example used Chapter 5 and 8) can teach students how to analyze these complex design problems. As students become comfortable with these ideas, it is my hope that they will develop their own unique approaches to solving design problems which will carry over into their engineering science course work. It is therefore important to teach students not only the tools from various different design approaches, but also the over-arching design methodology for solving problems. It is when students have this knowledge of an overarching design methodology that students can begin to select the design tools that are most appropriate to their problem.

To this end, the *framework* discussed in Chapter 8, is an attempt to extract to a level above the specific tools presented in this dissertation. This *framework* creates the underpinnings for a family of methodologies that may be used to identify automation opportunities. Using this *framework* as a starting point will allow methodologies consisting of other tools and methods to be used in conjunction with those presented in this dissertation to solve new and unconsidered automation problems.

10.3 Future Work

Future research will not only extend this research to further investigate automation opportunities, but will also investigate new directions in the fields of systems engineering and engineering design. In the following subsections a few of these potential avenues for future research are investigated.

10.3.1 Investigation of Automation Opportunities

Applications into the broader impact topics will be considered not only as case studies, but also as opportunities to extend the framework. In this research, error prone human-centric activities were considered as automation opportunities. But through these case studies, other automation opportunities are observed. Framework extensions might look for automation opportunities based on cost, productivity gain, process reliability, individual, social or global impacts and sustainable design. This list could continue as new applications will continue to be discovered, and as seen with manufacturing applications of automated systems, the implications of these applications will have to be learned.

10.3.2 Computational Underpinnings of Integrated Models

Integrated models containing functions, processes and environments will be explored as a starting point for a computational framework fusing function-based conceptual design tools. Initial research toward this computational framework has begun with the FunctionCAD application discussed in Chapter 4 (Nagel, Perry et al. 2009). In FunctionCAD, systems can be modeled at varying levels of fidelity and depth to capture environments, processes and functions. Models created in FunctionCAD are extensible and can be integrated with conceptual design tools. This connection, however, is a proof of concept requiring further development to connect models with the Design Repository (Design Engineering Lab 2008), behavioral models (Hutcheson et al. 2007), and visualization-based conceptual design tools (Bryant, McAdams et al. 2005; Bryant, Stone et al. 2005; Attaluri et al. 2006; Bryant et al. 2007) to create the integration illustrated in Figure 10.1.

The future integration of engineering design tools will allow design teams to actively update models. As components are selected from the concept generation tools, grammar rules, such as listed in Appendix A and B, would guide updates to the functional model based on the underlying theory of Form-Follows-Form (Bohm et al. 2009; Bohm and Stone 2010). Behavioral models would then perform back-of-the-envelope calculations based on the updated functionality and parametric equations stored in the design database. Models then actively validate during concept generation assisting with concept selection. The resulting computational tool will provide an automated environment where a design team can interactively develop a preliminary design, perform concept generation, validate concepts and archive designs.

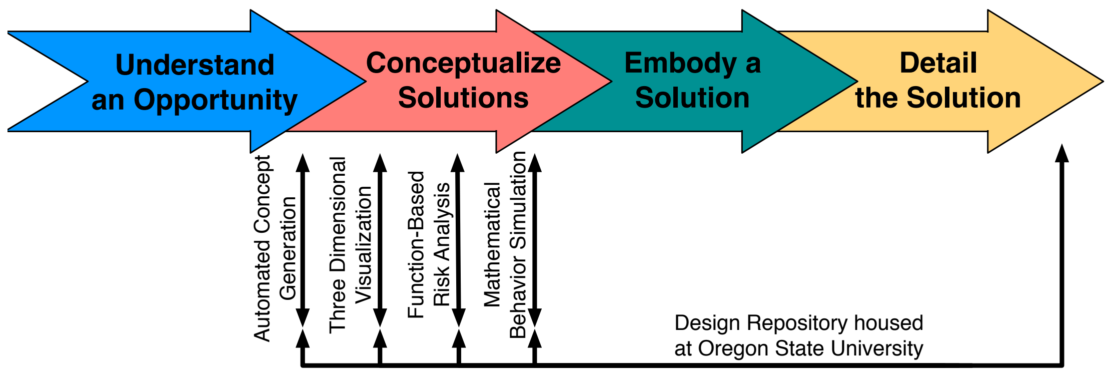


Figure 10.1 Mapping of concept generation tools to the high-level representation of the engineering design process.

10.3.3 Linking Process Failures to Functional Failures

Traditionally, failure analysis approaches focus on the components or functionality associated with a product and not on how the customer will use the product. In this dissertation, failures associated with how the customer interacts with a product are investigated. These process-based failures, however, have the potential to cause functional failures within the actual product. Future research will investigate the relationship between process-based failures and functional-failures. The resulting failure analysis approach will explore failure propagation based on integrated model flows. Failures will be traceable from the environment and processes to within the functionality of the product.

For example, consider again the Bosch CS20. Severing the power cable during the operation of the circular saw is a common failure reported by contractors over the life time of the circular saw. Once the power cable is severed, the saw is ruined. This failure occurring during the operation of the circular saw results in a rather obvious functional failure—the circular saw cannot turn on. Understanding not only the functional requirements, but also the process can help to lead to a solution for the failure. At first glance, the apparent solution to the failure might be to convert the circular saw to battery power. Batteries, however, do not provide the longevity required, so a power cable is required. Also, power cables are often used as a means to lower the circular saw to the ground when used from a ladder, thus freeing both hands for descent (Ulwick 2005). The solution adopted by Bosch was to make the cable removable; it can still be used to lower the saw to the ground, but if severed, it can be quickly replaced.

10.3.4 Sustainable Design Through Customer Cueing

As the popularity of more environmentally friendly products like hybrid automobiles indicate, customers are becoming more conscious of the challenges of sustainability. Engrained human actions, often however, make it difficult for consumers to accept new and different actions required by sustainable products. Future research using integrated models during preliminary design will investigate customer cueing to indicate more sustainable usage of products. Variables that are key to the customer's adoption of new environmentally friendly processes will be identified. These variables represent design opportunities where companies may cue customers of more sustainable product usage practices. Then during concept generation, processes or functionality associated with these cueing points will be used to identify strategies to cue customers of new methods required to maintain the anticipated sustainability for a product.

Consider, for example, the configuration model shown in Figure 10.2 abstracting the use of liquid laundry detergent. The model abstracts the operator being guided though the task of separating liquid detergent from the detergent bottle. The flow of detergent is directed by the flow rate control signal; the volume status signal indicates the quantity of detergent that has left the bottle and entered into the measuring device.

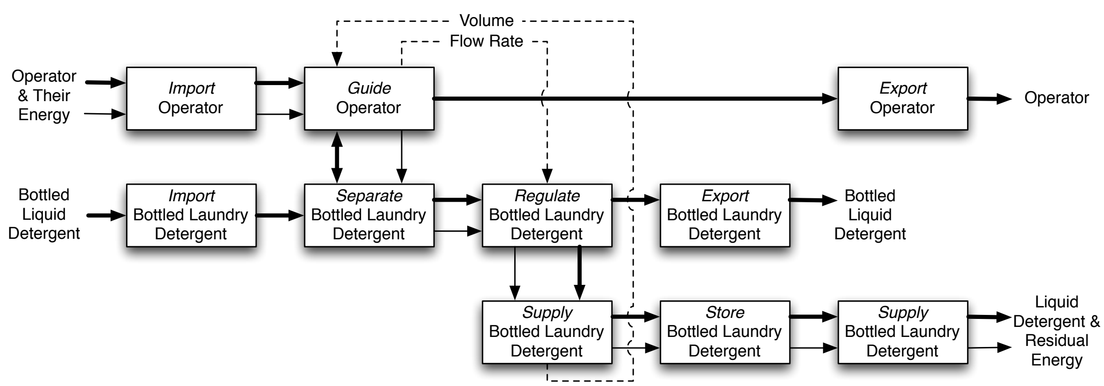


Figure 10.2 Configuration model abstracting the use of laundry detergent.

To cue the operator to use a more sustainable volume of laundry detergent, the two signal flows—one representing the volumetric flow rate and the other representing the volume poured—can be investigated. Investigation into these flows and their associated behavior—equations for volume and flow rate—reveals variables available for customer cueing. Volume (V), calculated per Equation (10.1) where r is the radius of the measuring apparatus and h is the depth of the measuring apparatus, has two variables, r and h, that can be adjusted by design modifications to cue the customer toward more sustainable usage practices. The volumetric flow rate, (Q), which is calculated per Equation (10.2) where A is the area of the exit orifice of the bottle and v is the flow velocity also has two potential variables, A and v. Adjusting r and h would require a modification of the detergent bottle cap either making the bottle shorter and/or narrower, while adjusting A and v would require modification of the bottle either by adjusting the handle to change the pour angle or by adjusting the spout to choke the flow rate.

$$V = \pi r^2 h \tag{10.1}$$

$$Q = Av \tag{10.2}$$

Procter and Gamble have used similar cueing ideas with the design of their new Tide liquid detergent bottle. The volume of the cap (by reducing *h)* is changed as well as the handle angle and location. These adjustments allow the user to get the same quantity of wash cycles from a single bottle of laundry detergent, while allowing the company to create a more sustainable product. With the new design, Procter and Gamble not only reduces the amount of water required in the product, but also reduces the amount of packing material to be recycled at the end of the product's life.

10.4 Closing Statement

Identifying actions ripe for automation can take many avenues, and while it was necessary to scope this problem for this dissertation, I believe that each of these avenues are equally important. As automated systems increasingly become a part of our lives, we will have to answer the tough questions of when to automate and when not to. The research presented in this dissertation is a starting point to developing answers to those questions.

BIBLIOGRAPHY

(2001). "Abstraction." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, Inc, 8.

(2001). "Automation." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, Inc, 141.

(2001). "Environment." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, Inc, 650.

(2001). "Framework." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, Inc, 760.

(2001). "Method." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, 1209.

(2001). "Methodology." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, Inc, 1209.

(2001). "Model." *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, NY, Random House Value Publishing, Inc, 1235-6.

(2005). "Abstraction." *The New Oxford American Dictionary*. Oxford, Oxford University Press, Inc.

(2005). "Grammar." *The New Oxford American Dictionary*. Oxford, Oxford University Press, Inc.

(December 2004). "Best of What's New 2004—Home Tech." *Popular Science* 265(6).

Altshuller, G. (1995). *Creativity As An Exact Science: The Theory of the Solution of Inventive Problems*. Luxembourg, Gorden and Breach Science Publishers Inc.

Altshuller, G. (2005). *40 Principles (Extended Edition): TRIZ Keys to Technical Innovation*. Worcester, MA, Technical Innovation Center, Inc.

Ambler, S. W. (2009). "UML 2 Use Case Diagrams." Retrieved April 15, 2010, from http://www.agilemodeling.com/artifacts/useCaseDiagram.htm.

Andersson, J., J. Pohl and S. D. Eppinger (1998). "A Design Process Modeling Approach Incorporating Nonlinear Elements." *ASME Design Theory and Methodology Conference*, Atlanta, GA, ASME. DTM-5663.

Apple Corporation (2004). "iSight User's Guide." http://manuals.info.apple.com/en_US/iSightUserGuide.pdf.

Asimow, M. (1962). *Introduction to Design*. Englewood Cliffs, NJ, Prentice-Hall.

Atman, C. J. and K. M. Bursic (1996). "Teaching Engineering Design: Can Reading a Textbook Make a Difference?" *Research in Engineering Design* 8(4): 240-250.

Attaluri, V., D. A. McAdams, R. B. Stone and A. D. Crescenzo (2006). "Visual Representations as an Aid to Concept Generation." *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Philadelphia, PA, ASME. DETC2006-99572.

Automotive Industry Action Group (AIAG) (1993). "Potential Failure Mode and Effects Analysis (FMEA) Reference Manual." Automotive Industry Action Group.

Bedford, T. and R. Cooke (2001). *Probabilistic Risk Analysis: Foundations and Methods*. Cambridge, Cambridge University Press.

Blanchard, B. S. and W. J. Fabrycky (2006). *Systems Engineering and Analysis*. Upper Saddle River, NJ, Prentice-Hall.

Bohm, M. and R. Stone (2004). "Representing Product Functionality to Support Reuse: Conceptual and Supporting Functions." *Proceedings of DETC2004*, Salt Lake City, UT. DETC2004-57693.

Bohm, M., R. Stone and R. Nagel (2009). "Form Follows Form - Is a New Paradigm Needed?" *ASME International Mechanical Engineering Congress*, Lake Buena Vista, FL, ASME. IMECE2009-10410.

Bohm, M. R. and R. B. Stone (2010). "Form Follows Form - Fine Tuning Artificial Intelligence Methods." *ASME International Design Engineering Technical Conferences*, Montreal, Quebec, Canada, ASME. DETC2010-28774.

Bohm, M. R., J. P. Vucovich and R. B. Stone (2008). "Using a Design Repository to Drive Concept Generation." *Journal of Computer and Information Science in Engineering* 8(1).

Bosch. "7-1/4" Circular Saw with Direct Connect / Model: CS20." Retrieved Jan. 20, 2010, from http://boschtools.com/products/tools/pages/boschproductdetail.aspx?pid=CS20.

Bowman, R. A. (2007). "Efficient sensitivity analysis of PERT network performance measures to significant changes in activity time parameters." *Journal of Operational Research Society* 58: 1354-1360.

Bradley, D. A., D. Dawson, N. C. Burd and A. J. Loader (1991). *Mechatronics: Electronics in Products and Processes*. London, Chapman and Hall.

Brown, A. S. (2010). "The Drone Warriors." *Mechanical Engineering* 132(1): 22-27.

Bryant, C., D. McAdams, R. Stone, T. Kurtoglu and M. Campbell (2005). "A Computational Technique for Concept Generation." *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA, ASME. DETC2005-85323.

Bryant, C., R. Stone, D. McAdams, T. Kurtoglu and M. Campbell (2005). "Concept Generation from the Functional Basis of Design." *International Conference on Engineering Design*, Melbourne, Australia.

Bryant, C. R., M. R. Bohm, R. B. Stone and D. A. McAdams (2007). "An Interactive Morphological Matrix Computational Design Tool: A Hybrid of Two Methods." *ASME International Design Engineering Technical Conferences*, Las Vegas, NV, ASME. DETC2007-35583.

Bryant, C. R., R. B. Stone and D. A. McAdams (2006). "Automated Concept Generation from the Functional Basis of Design." *Accepted in Research in Engineering Design January 2007*.

Cagan, J. and C. M. Vogel (2002). *Creating Breakthrough Products: Innovation from Product Planning to Program Approval*. Upper Saddle River, NJ, Prentice Hal, Inc.

Chadia, A., D. Maloney-Krichmar and J. Preece (2004). "User-Centered Design." *Berkshire Encyclopedia of Human-Computer Interaction*. W. S. Bainbridge. Great Barrington, MA, Berkshire Publishing Group LLC, 763-767.

Chandrasekaran, B. (1994). "Function representation and causal processes." *Advances in Computers.* 38: 73-143.

Chen, L., M. Jayaram and J. F. Xi (2002). "A New Functional Representation Scheme for Conceptual Modeling of Mechatronic Systems." *Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, Montreal, Canada, ASME.

Cho, S.-H. and S. D. Eppinger (2001). "Product Development Process Modeling Using Advanced Simulation." *ASME International Design Engineering Technical Conferences*, Pittsburgh, PA, ASME. DTM-21691.

Christensen, C. M. and M. E. Raynor (2003). *The Innovator's Solution: Generating and Sustaining Successful Growth*. Boston, Harvard Business School Press.

Cross, N. (2000). *Engineering Design Methods: Strategies for Product Design*. Chichester, John Wiley & Sons, LTD.

Cutherell, D. (1996). *Chapter 16: Product Architecture*, Wiley and Sons.

De Silva, C. W. (2005). *Mechatronics-An Integrated Approach*. Boca Raton, CRC Press.

Defense Science Board Task Force (Sept 2003). "The Role and Status of DoD Red Teaming Activities." Office of the Under Secretary of Defense.

Design Engineering Lab. (2008). "The Design Repository @ Oregon State University." Retrieved Oct. 17, 2009, from repository.designengineeringlab.org.

Dieter, G. (1991). *Engineering Design: A Materials and Processing Approach*. New York, McGraw-Hill.

Dutson, A. J., R. H. Todd, S. P. Magleby and C. D. Sorensen (1997). "A Review of Literature on Teaching Engineering Design Through Project-oriented Capstone Courses." *Journal of Engineering Education* 86(1): 17-28.

Dym, C. L. (1999). "Learning Engineering: Design, Languages, and Experiences." *Journal of Engineering Education* 88(2): 145-148.

Dym, C. L., A. M. Agogino, O. Eris, D. D. Frey and L. J. Leifer (2005). "Engineering Design Thinking, Teaching, and Learning." *Journal of Engineering Education* 94(1): 103-120.

Dym, C. L. and P. Little (2004). *Engineering Design: A Project-based Introduction*, John Wiley & Sons, INC.

Earle, J. H. (1990). *Engineering Design Graphics*. Reading, Mass, Addison Wesley.

Eder, J. and W. Liebhard (1995). "The Workflow Activity Model WAMO." *International Conference on Cooperative Information Systems (CoopIS)*, Vienna, Austria.

Eder, W. E. and S. Hosnedl (2008). *Design Engineering: A Manual for Enhanced Creativity*. Boca Raton, CRC Press, Taylor & Francis Group.

Erden, M. S., H. Komoto, T. J. Van Beek, V. D'Amelio, E. Echavarria and T. Tomiyama (2008). "A review of functional modeling: Approaches and applications." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22(2): 147-169.

Eubanks, C. F., S. Kmenta and K. Ishii (1997). "Advanced failure modes and effects analysis using behavior modeling." *ASME Design Engineering Technical Conferences*, Sacramento, CA. DETC97/DTM-3872.

Fagerström, B. and P. Nilsson (2003). "Integrated Product and Process Modeling." *ASME Design Engineering Technical Conferences*, Chicago, IL, ASME. DTM-48656.

Friedenthal, S., A. Moore and R. Steiner (2008). *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann.

Gausemeier, J., M. Flath and S. Möhringer (2001). "Conceptual Design of Mechatronic Systems Supported by Semi-formal Specification." *IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings*, Como, Italy, IEEE/ASME.

Gero, J. S. (1990). "Design Prototypes: A Knowledge Representation Schema for Design." *AI Magazine* 11(4): 26-36.

Gero, J. S. and U. Kannengiesser (2002). "The Situated Function - Behaviour - Structure Framework." *Artificial Intelligence in Design*: 89-104.

Gibson, J. J. (1986). "Chapter 8: The Theory of Affordances." *The Ecological Approach to Visual Perception*. Hillsdale, NJ, Lawrence Erlbaum Associates, Inc.: 127-143.

Goel, A. K. and B. Chandrasekaran (1992). "Case-based design: a task analysis." *Artificial Intelligence Approaches to Engineering Design* 2: 165-184.

Goel, A. K., S. Rugaber and S. Vattam (2009). "Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23(1): 23-35.

Griffin, A. and J. R. Hauser (1993). "The Voice of the Customer." *Marketing Science* 12(1): 1-27.

Hauser, J. and D. Clausing (1988). "The House of Quality." *Harvard Business Review* 66(3): 63-73.

Hill, P. H. (1970). *The Science of Engineering Design*. New York, Holt, Rinehart and Winston.

Hirtz, J., R. Stone, D. McAdams, S. Szykman and K. Wood (2002). "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts." *Research in Engineering Design* 13(2): 65-82.

Hokuyo Automatic Co., L. (2005). "Range-Finder Type Laser Scanner URG-04LX Specifications."
http://www.acroname.com/robotics/parts/R283-HOKUYO-LASER1s.pdf.

Hundal, M. (1990). "A Systematic Method for Developing Function Structures, Solutions and Concept Variants." *Mechanism and Machine Theory* 25(3): 243-256.

Husqvarna. (2009). "Robotic Lawn Mowers for Homeowners | Husqvarna Automower." Retrieved April 1, 2010, from http://www.husqvarna.com/us/homeowner/products/robotic-mowers/husqvarna-robotic-mowers-for-homeowners/.

Hutcheson, R. S., D. A. McAdams, R. B. Stone and I. Y. Tumer (2007). "Function-Based Behavioral Modeling." *ASME International Design Engineering Technical Conferences*, Las Vegas, NV, ASME. DETC2007-35337.

Hutcheson, R. S., D. A. McAdams, R. B. Stone and I. Y. Tumer (2007). "Function-Based Behavioral Modeling." *ASME International Design Engineering Technical Conferences*, Las Vegas, NV, ASME. DETC2007-35337.

Hyman, B. (1998). *Fundamentals of Engineering Design*. Upper Saddle River, New Jersey, Prentice Hall.

Intelligent Ground Vehicle Competition. (2008). "The 16th Annual Intelligent Ground Vehicle Competition." Retrieved April, 2008, from www.IGVC.org.

iRobot. (2010). "iRobot: Cleaning Robots." Retrieved April 2, 2010, from http://store.irobot.com/shop/index.jsp?categoryId=2804605.

Jayaram, M., L. Chen and J. F. Xi (2003). "Functional Modeling of Complex Mechatronic Systems." *Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, Chicago, Il, ASME.

Jensen, D., I. Y. Tumer and T. Kurtoglu (2009). "Flow State Logic (FSL) for Analysis of Failure Propagation in Early Design." *ASME International Design Engineering Technical Conferences*, San Diego, CA, ASME. DETC2009-87064.

Jones, J. P. (1976). *Gathering and Writing the News*. Chicago, Nelson-Hall.

Ju, T. L. (2001). "Process Definition for Adaptable Workflow Management Systems." *XML Europe Conference*, Berlin, Germany, Internationales Congress Centrum (ICC).

Katz, G. M. (2001). "The "One Right Way" to Gather the Voice of the Customer." *PDMA Visions Magazine* 25(4).

Kelley, J. and M. Walker (1959). "Critical-Path Planning and Scheduling." *Proceedings of Eastern Joint Computer Conference*.

Kmenta, S., P. Fitch and K. Ishii (1999). "Advanced Failure Modes and Effects Analysis of Complex Processes." *ASME Design Engineering Technical Conferences*, Las Vegas, NV. DETC99/DFM-8939.

Kmenta, S. and K. Ishii (1998). "Advanced FMEA using Meta Behavior Modeling for Concurrent Design of Products and Controls." *ASME Design Engineering Technical Conferences*, Atlanta, GA, ASME. DETC98/CIE-5702.

Koller, R. (1985). *Konstruktionslehre für den Maschinenbau (Mechanical Engineering Design)*. Berlin: Springer-Verlag.

Krus, D. and K. Grantham Lough (2007). "Applying Function-Based Failure Propagation in Conceptual Design." *ASME International Design Engineering Technical Conferences*, Las Vegas, NV.

Kumamoto, H. and E. J. Henley (1996). *Probabilistic Risk Assessment and Management for Engineers and Scientists*. New York, IEEE Press.

Kurtoglu, T. and M. I. Campbell (2009). "Automated Synthesis of Electromechanical Design Configurations from Empirical Analysis of Function to Form Mapping." *Journal of Engineering Design* 20(1): 83-104.

Kurtoglu, T. and I. Y. Tumer (2007). "A Graph-Based Framework for Early Assessment of Functional Failures in Complex Systems." *ASME International Design Engineering Technical Conferences*, Las Vegas, NV. DETC2007-35421.

Kurtoglu, T. and I. Y. Tumer (2007). "FFIP: A Framework for Early Assessment of Functional Failures in Complex Systems." *International Conference on Engineering Design*, Paris, France. ICED'07/479.

Kyodo America. (2010). "LawnBott." Retrieved April 1, 2010, from http://www.lawnbott.com/.

Leonard, D. (2002). "The Limitations on Listening." *Harvard Business Review* 80(1): 93.

Lindbeck (1994). *Product Design and Manufacturing*. Englewood Cliffs, NJ, Prentice Hall.

Little, A., K. Wood and D. McAdams (1997). "Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking and Redesign." *Proceedings of the 1997 Design Engineering Technical Conferences*, Sacramento, CA. 97-DETC/DTM-3879.

Maier, J. R. A. and G. M. Fadel (2003). "Affordance-based Methods for Design." *ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, IL, ASME. DETC2003/DTM-48673.

Malcolm, D. G., J. H. Roseboom, C. E. Clark and W. Fazar (1959). "Application of a Technique for Research and Development Program Evaluation." *Operations Research* 7: 646-669.

Marshall, C. W. (1971). *Applied Graph Theory*. New York, Wiley-Interscience.

Mayer, R. J., C. P. Menzel, M. K. Painter, P. S. deWitte, T. Blinn and B. Perakath (1995). "Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report." Knowledge Based Systems, Incorporated.

Middleton, S. R. (2000). "Developing an Automation Concept That Is for Your Laboratory." *Clinical Chemistry Forum* 46(No. 5): 757-763.

MIL-STD-1629A (1980). "Procedures for Performing a Failure Mode, Effects and Criticality Analysis." Department of Defense.

Miles, L. (1961). *Techniques of Value Analysis and Engineering*. New York, McGraw-Hill.

Millward, C. M. (1996). *A Biography of the English Language*. Fort Worth, TX, Harcourt Brace College Publishers.

Nagel, R., J. L. Greer, R. B. Stone and D. A. McAdams (2009). "Fault Propagation and Sensitivity Analysis to Support Counterterrorism Activities." *19th Annual INCOSE International Symposium*, Singapore, INCOSE. Paper No. 156.

Nagel, R., K. Perry, R. B. Stone and D. A. McAdams (2009). "FunctionCAD: A Functional Modeling Application Based on the Function Design Framework." *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, San Diego, CA, ASME. DETC2009-87010.

Nagel, R. L., M. R. Bohm, R. B. Stone and D. A. McAdams (2007). "A Representation of Carrier Flows for Functional Design." *International Conference on Engineering Design*, Paris, France.

Nagel, R. L., R. S. Hutcheson, R. Stone, D. McAdams and J. Donndelinger (2008). "Function Design Framework (FDF): Integrated Process and Function Modeling for Complex System Design." *ASME International Design Engineering Technical Conferences*, Brooklyn, NY, ASME. DETC2008-49369.

Nagel, R. L., R. S. Hutcheson, R. B. Stone and D. A. McAdams (2009). "Process and Event Modelling for Conceptual Design." *Journal of Engineering Design*.

Nagel, R. L., R. B. Stone and D. A. McAdams (2006). "A Process Modeling Methodology for Automation of Manual and Time Dependent Processes." *ASME International Design Engineering Technical Conferences*, Philadelphia, PA. DETC2006-99437.

Nagel, R. L., J. P. Vucovich, R. B. Stone and D. A. McAdams (2007). "Signal Flow Grammar From the Functional Basis." *International Conference on Engineering Design*, Paris, France. 633.

Nagel, R. L., J. P. Vucovich, R. B. Stone and D. A. McAdams (2008). "A Signal Grammar to Guide Functional Modeling of Electromechanical Products." *Journal of Mechanical Design* 130(5): 136.

National Institute of Standards and Technology (1993). *Integration Definition for Function Modeling (IDEF0)*, Information Processing Standards Publications (FIPS PUBS).

Niku, S. B. (2009). *Creative Design of Products and Systems*. Hoboken, NJ, John Wiley & Sons, Inc.

Norman, D. A. (2002). *The Design of Everyday Things*. New York, Basic Books.

Olewnik, A. and K. Lewis (2007). "Limitations of the House of Quality to provide quantitative design information." *International Journal of Quality and Reliability Management* 25(2): 125-146.

Otto, K. and K. Wood (2001). *Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development*. New York, Prentice-Hall.

Padalkar, S., G. Karsai, C. Biegl, J. Sztipanovits, K. Okuda and N. Miyasaka (1991). "Real-Time Fault Diagnostics." *IEEE Expert: Intelligent Systems and Their Applications* 6(3): 75-85.

Pahl, G. and W. Beitz (1984). *Engineering Design: A Systematic Approach*. London, Design Council.

Pahl, G., W. Beitz, J. Feldhusen and K. H. Grote (2007). *Engineering Design: A Systematic Approach*, Springer Verlag.

Pearlstein, S. (2010). "End of Saturday mail service is only the beginning." *Washington Post*. Washington DC.

Pethokoukis, J. M. (2004). "Meet Your New Coworker: Industrial robots are reshaping manufacturing." *US New & World Report*. March 15.

Popovic, D. and L. Vlacic, Eds. (1999). *Mechatronics in Engineering Design and Product Development*. New York, Marcel Dekker.

Pugh, S. (1991). *Total Design: Integrated Methods for Successful Product Engineering*. Wokingham, England, Addison-Wesley Publishing Company.

Quirk, R., S. Greenbaum, G. Leech and J. Svartvik (1985). *A Comprehensive Grammar of the English Language*. London, Longman.

Rodenacker, W. (1971). *Methodisches Konstruieren (Methodical Design)*. Berlin, Heidelberg, New York, Springer.

Roth, K. H. (1981). "Foundation of Methodical Procedures in Design." *Design Studies* 2(2): 107-115.

Russomanno, D. J., R. D. Bonnell and J. B. Bowles (1993). "Functional Reasoning in Failure Modes and Effects Analysis (FMEA) Expert System." *Reliability and Maintainability Symposium*.

Schmitz, J. and M. Manceor (2009). "Door Assist System Allows Easier Entry Into Heavily Armored Vehicles." *Accelerate*(3): 58-61.

Schoenherr, E. (2009). "Moving Future Convoy Operations with Convoy Active Safety Technologies (CAST)." *Accelerate*(3): 74-79.

Schroeder, F. E. H. (1993). *Front Yard America: The Evolution and Meanings of a Vernacular Domestic Landscape*. Bowling Green, OH, Bowling Green State University Popular Press.

Schultz, W. (1999). *A Man's Turf: The Perfect Lawn*. New York, Clarkson N. Potter, Inc.

Sheridan, T. B. (2002). *Humans and Automation: System Design and Research Issues*. Santa Monica, CA, A John Wiley & Sons, Inc.

Stamatelatos, M. (2000). "Probabilistic Risk Assessment: What is it and Why is it Worth it?". NASA Office of Safety and Mission Assurance.

Stamatis, D. H. (1995). *Failure Mode and Effect Analysis, FMEA from Theory to Execution*. ASQ Quality Press, Milwaukee, WI.

Stank, T. P., P. J. Daugherty and A. E. Ellinger (1997). "Voice of the Customer: The Impact on Customer Satisfaction." *International Journal of Purchasing and Materials Management* 33(4): 2-9.

Stone, R., N. Hubing, R. Fannin and R. Flori (2005). "The Interdisciplinary Design Engineering Department: A Systems View, A Design Focus and Customizable Interdisciplinary Tracks." *Proceedings of IDETC/CIE 2005*, Long Beach, CA, ASME. DETC2005-84874.

Stone, R. and K. Wood (2000). "Development of a Functional Basis for Design." *Journal of Mechanical Design* 122(4): 359-370.

Stone, R., K. Wood and R. Crawford (2000). "Using Quantitative Functional Models to Develop Product Architectures." *Design Studies* 21(3): 239-260.

Stone, R. B., I. Y. Tumer and M. Van Wie (2005). "The Function-Failure Design Method." *Journal of Mechanical Design* 127(3): 397-407.

Suh, N. (1990). *The Principles of Design*. New York, NY, Oxford University Press, Inc.

Suh, N. P. (2001). *Axiomatic Design: Advances and Applications*. New York, Oxford University Press.

SYSML.org. (2008). "SysML - Open Source Specification Project." Retrieved April 15, 2010, from http://www.sysml.org/.

Szykman, S., J. Racz and R. Sriram (1999). "The Representation of Function in Computer-Based Design." *ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Las Vegas, NV. DETC99/DTM-8742.

The Object Management Group. (2009). "Unified Modeling Language - UML." Retrieved Feb. 3, 2009, from http://www.uml.org/.

Theisen, B. L. and P. Richardson (2009). "Construction Engineering Robot Kit: Warfighter Experiment." *Intelligent Robots and Computer Vision XXVI: Algorithms and Techniques*, San Jose, CA, SPIE. 7252.

Tor, S. B., G. A. Britton, W. Y. Zhang and Y. M. Deng (2002). "Guiding Functional Design Through Rule-based Causal Behavior Reasoning." *International Journal of Production Research* 40(3): 667-682.

U.S. National Library of Medicine and National Institutes of Health. (2010). "Assistive Devices: MedlinePlus." Retrieved April 1, 2010, from http://www.nlm.nih.gov/medlineplus/assistivedevices.html#cat59.

Ullman, D. G. (2010). *The Mechanical Design Process*. Boston, McGraw-Hill.

Ulrich, K. T. and S. D. Eppinger (2004). *Product Design and Development*. Boston, MA, McGraw-Hill/Irwin.

Ulwick, A. W. (2002). "Turn Customer Input into Innovation." *Harvard Business Review* 80(1): 91-97

Ulwick, A. W. (2005). *What Customers Want: Using Outcome-Driven Innovation to Create Breakthrough Products and Services*. New York, McGraw-Hill.

Umeda, Y., H. Takeda, T. Tomiyama and H. Yoshikawa (1990). "Function, Behaviour, and Structure." *AIENG'90 Applications of AI in Engineering*: 177-193.

Vesely, W. E. and F. F. Goldberg (1981). "Fault Tree Handbook." US Nuclear Regulatory Commission. NUREG-0492.

Voland, G. (2004). *Engineering By Design*. Upper Saddle River, NJ, Pearson Prentice Hall.

Welch, R. V. and J. R. Dixon (1992). "Representing Function, Behavior and Structure during Conceptual Design." *4th International Conference on Design Theory and Methodology*, Scottsdale, AZ, ASME.

Wilkinson, P. J. and T. P. Kelly (1998). "Functional Hazard Analysis for Highly Integrated Aerospace Systems." *Certification of Ground/Air Systems Seminar*. Ref. No. 1998/255.

Williams, C. (2009). "Sensor-Enhanced Armor May Be Key to Better Vehicle Protection." *Accelerate*(3): 66-69.

Zhang, W. Y., S. B. Tor, G. A. Britton and Y. M. Deng (2002). "Functional Design of Mechanical Products Based on Behavior-Driven Function-Environment-Structure Modeling Framework." *Innovation in Manufacturing Systems (IMST) @ http://hdl.handle.net/1721.1/4031*.

Zwicky, F. (1969). *Discovery, Invention, Research - Through the Morphological Approach*. Toronto, The Macmillian Company.

APPENDICES

A Functional Morphology

The following morphology rules are meant to guide the assembly of functional models following the Functional Basis. Each rule describes how a function is to be used in a functional model and the flows with which it is compatible. Images are used to illustrate each of the morphology rules.

Rule 1: Use the *branch* functions to represent the disconnection of a single (EMS) flow into two or more (EMS) flows.

- All *branch* functions require a single flow entering the function block and two or more flows exiting.

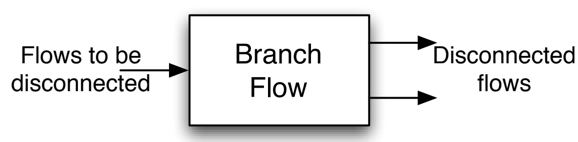


Figure A.1 Functional morphology for branch flows functionality.

a. Use the *separate* functions to **branch** a single (EMS) flow into multiple flows.

   - Each output flow from *separate* functions are distinct from other output flows and the input flow.

   i. Use *divide* to **separate** a single (EMS) flow of mixed elements into multiple flows of sorted elements.

   ii. Use *extract* to **separate** non-homogenous (EMS) flows from an otherwise homogenous medium.

   iii. Use *remove* to **separate** a part of a homogenous medium as a (EMS) flow.

b. Use *distribute* to **branch** a single (EMS) flow into multiple (EMS) flows.

   - Output flows from *distribute* are neither distinct from each other nor from the input flow.

Rule 2: Use the *channel* functions to represent the motion of a (EMS) flow from one location to another.

- Flows entering *channel* functions match those exiting.

c. Use *import* to **channel** a (EMS) flow from the outside of a system boundary to the inside of a system boundary.

   - Flow arrows transverse the system boundary to an *input* function block representing flow into the system.

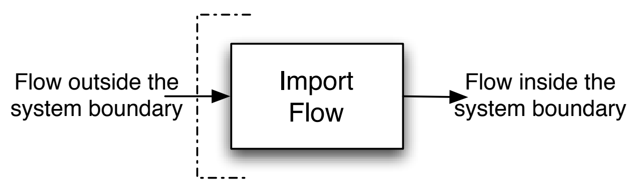


Figure A.2 Functional morphology for import flow functionality.

d. Use *export* to **channel** a (EMS) flow from the inside of a system boundary to the outside of a system boundary.

   - Flow arrows transverse the system boundary from an *export* function block representing flow from the system.

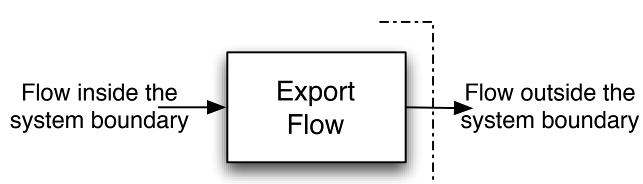


Figure A.3 Functional morphology for export flow functionality.

e. Use the *transfer* functions to **channel** a (EMS) flow from one place to another along an unspecified and nonspecific route.

  i. Use *transport* to **transfer** material flows.

  - *Transport* signal flows with material flow carriers.

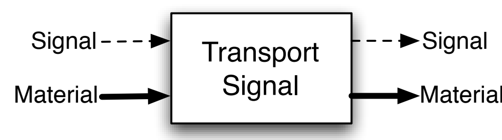


Figure A.4 Functional morphology for transport signal functionality.

  ii. Use *transmit* to **transfer** energy flows.

  - *Transmit* signal flows with energy flow carriers.

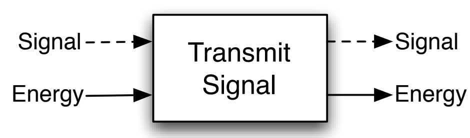


Figure A.5 Functional morphology for transmit signal functionality.

f. Use the *guide* functions to **channel** a (EMS) flow along a predefined and rigid route.

  i. Use *translate* to **guide** a (EMS) flow in a single linear direction.

  ii. Use *rotate* to **guide** a (EMS) flow in around a single axis.

  iii. Use *allow DOF* to **guide** a (EMS) flow along a specified path via an applied energy or material flow.

  - *Allow DOF* functions require either an energy or material flow to enforce flow path.

- For conservation of flows, the energy or material flow required to enforce flow path is drawn leaving the *allow DOF* function block.

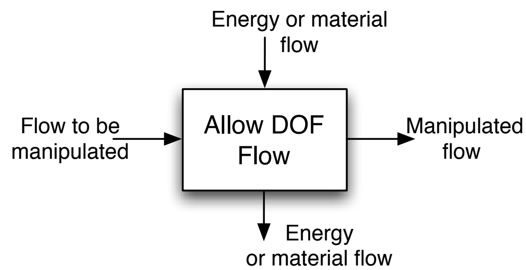


Figure A.6 Functional morphology for allow DOF flow functionality.

Rule 3: Use the *connect* functions to represent the merging of two or more (EMS) flows into a single (EMS) flow.

- All *connect* functions require two or more flows entering the function block and a single flow exiting.

a. Use the *couple* functions to **connect** two or more (EMS) flows such that the constituents form a single non-homogenous flow.

   i. Use *join* to **couple** two or more (EMS) flows in a predetermined manner.

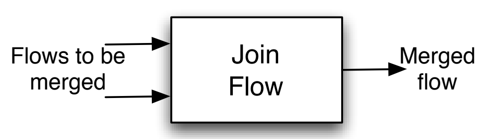


Figure A.7 Functional morphology for join flow functionality.

   ii. Use *link* to **couple** two or more (EMS) flows in a predetermined manner by means of an intermediary.

- *Link* functions require either an energy or material flow to affect the flow merging.
- The intermediary energy or material flow becomes a part of the single merged flow.

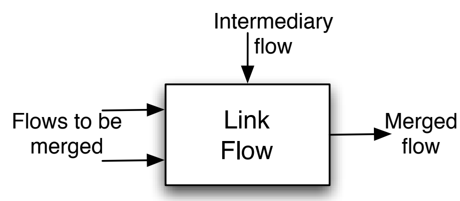


Figure A.8 Functional morphology for link flow functionality.

b. Use the *mix* function to **connect** two or more (EMS) flows such that the constituents form a single homogeneous flow.

Rule 4: Use the *control magnitude* functions to represent an adjustment in the size or amplitude of a flow.

a. Use the *actuate* function to **control magnitude** through a discrete toggling of a (EMS) flow.
   - *Actuate* functions require a *discrete control signal* flow to toggle state.

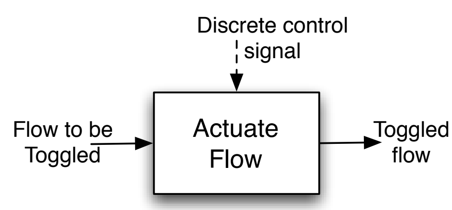


Figure A.9 Functional morphology for actuate flow functionality.

b. Use the *regulate* function to **control magnitude** of a (EMS) flow quantity in a specified analog manner.

- *Regulate* functions require an *analog control signa*l flow to adjust flow quantity.

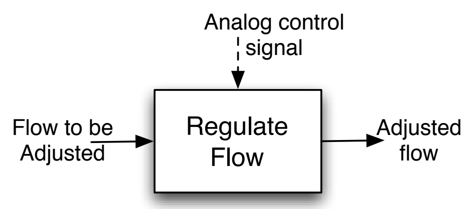


Figure A.10 Functional morphology for regulate flow functionality.

i. Use *increase* to **regulate** the enlargement of (EMS) flow magnitude.

ii. Use *decrease* to **regulate** the reduction of (EMS) flow magnitude.

c. Use the *change* function to **control magnitude** of a (EMS) flow in a predetermined and fixed manner.

- Change functions do not employ a *control signal* flow to adjust flow quantity.

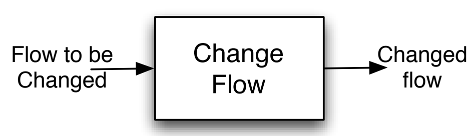


Figure A.11 Functional morphology for change flow functionality.

i. Use *increment* to affect **change** in a (EMS) flow resulting in a predetermined enlargement of flow magnitude.

ii. Use *decrement* to affect **change** in a (EMS) flow resulting in a predetermined reduction of flow magnitude.

iii. Use *shape* to affect **change** to the physical form of a material flow in a predetermined manner.

- *Shape* signal flows with material flow carriers.

iv. Use *condition* to affect **change** to an energy flow to render the flow appropriate for the desired use.

- *Condition* signal flows with energy flow carriers.

d. Use the *stop* function to **control magnitude** of a (EMS) flow terminally.

- For conservation of flows, a flow that enters a *stop* function block must also exit; the flow, however, may take a different form, be greatly reduced in magnitude or may directly leave the system as if the *stop* function were an *export* function.

i. Use *prevent* to **stop** a (EMS) flow from acting on the system.

- A flow entering a *prevent* function block no longer acts on the system. The flow leaves function block and the system.
- The flow may leave the function block in a different form than it entered.

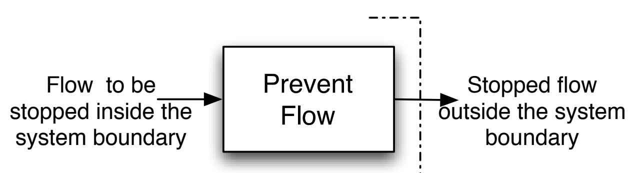


Figure A.12 Functional morphology for prevent flow functionality.

ii. Use *inhibit* to **stop** a (EMS) flow in a restrictive manner.

- A flow entering a *inhibit* function block is lessened and can still act on the system. The flow does not leave the system.
- The flow may leave the function block in a different form than it entered.

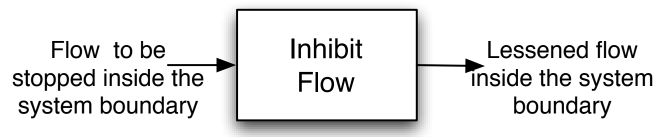


Figure A.13 Functional morphology for inhibit flow functionality.

Rule 5: Use the *convert* function to perform the conscious act of changing a flow from type (EMS) to another (EMS).

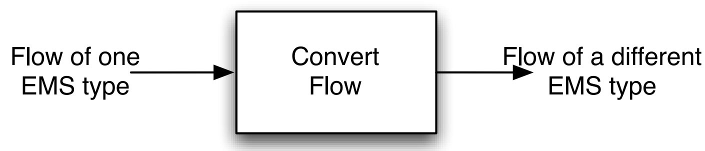


Figure A.14 Functional morphology for convert flow functionality.

Rule 6: Use the *provide* functions when (EMS) flows must be accumulated and/or dispensed.

- *Provide* functions tend to occur in pairs. If flows that are accumulated within the system are to later be used by the system they must be provided out of their accumulated state.

a. Use the *store* function when **providing** for the accumulation of a (EMS) flow.

b. Use the *supply* function when **providing** an (EMS) flow from its accumulated state.

   - When flows are accumulated within a system using a *store* function, a *supply* function is required to provide flows from their accumulated state.

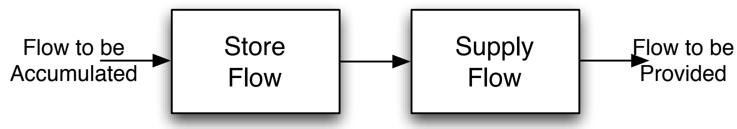


Figure A.15 Functional morphology for provision flow functionality.

Rule 7: Use the *signal* functions when information must be provided, processed or received about a (EMS) flow.

- Information is supplied in the form of a *signal* flow.
- Information about a (EMS) flow can be supplied either internally to the system or externally to the user.

a. Use the *sense* function to **signal** information concerning the detection or measurement of a (EMS) flow to the system.

- *Sense* functions require an input of the flow of interest to output a *status signal* flow representing data collected.
- The (EMS) flow, from which information is being collected, passes through the function block unchanged.

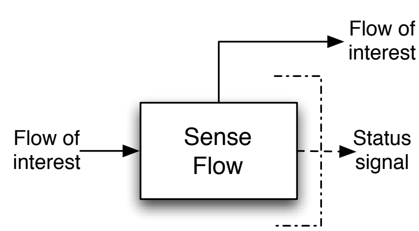


Figure A.16 Functional morphology for sense flow functionality.

i. Use *detect* when a discovery of information about a (EMS) flow or the presence of a (EMS) is to be **sensed**.

ii. Use *measure* when a (EMS) flow magnitude is to be **sensed**.

b. Use the *indicate* function to **signal** information about the status of the system to the user.

- *Indicate* functions end flow paths; thus, *status signal* flows exiting an *indicate* function block leave the system and do not connect to other function blocks.

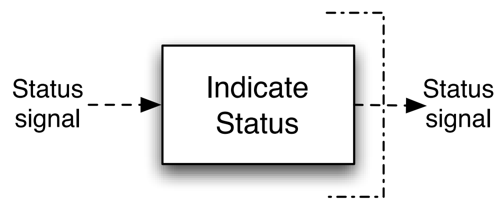


Figure A.17 Functional morphology for indicate status functionality.

i. Use *track* to **indicate** dynamic system information to the user.

ii. Use *display* to **indicate** static system information to the user.

c. Use the *process* function to **signal** the execution of a series of operations to extract conditional information on a *signal* flow.

- Either control or status flows can enter *process* function blocks; however, respective entering flows must also exit.

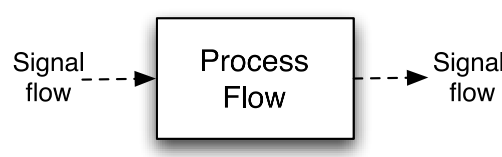


Figure A.18 Functional morphology for process flow functionality.

Rule 8: Use the *support* functions when a material flow is to be fixed firmly in a specified location.

a. Use the *stabilize* function to **support** a material flow such that it is placed firmly and unlikely to shift from a specified location.

b. Use the *secure* function to **support** a material flow such that it is firmly fixed in a specified location.

c. Use the *position* function to **support** a material flow into a specified orientation allowing action to be taken on the material flow.

B Functional Syntax

The following syntax are based on the functional morphology. Their purpose is to illustrate how the functional morphology are used in an actual function chain. Syntax flows are marked with nodes as black dots to illustrate where each "chunk" connects with other elements of a function chain. System boundaries are marked with brackets over flow lines. Where limited flow types are required, text is added for clarification.

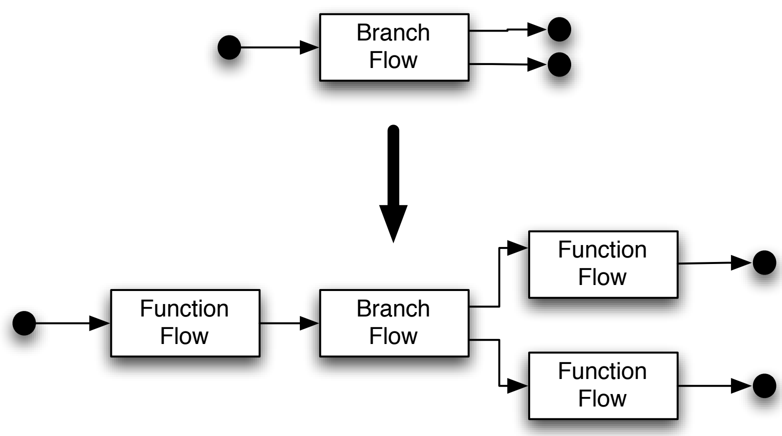


Figure B.1 Syntax number one, divider.

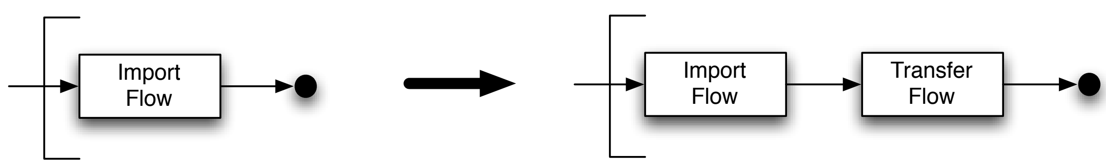


Figure B.2 Syntax number two, receiver.

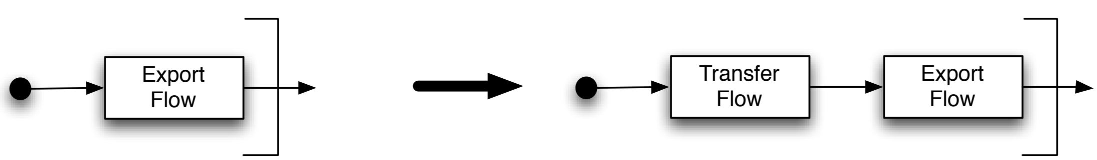


Figure B.3 Syntax number three, emitter.

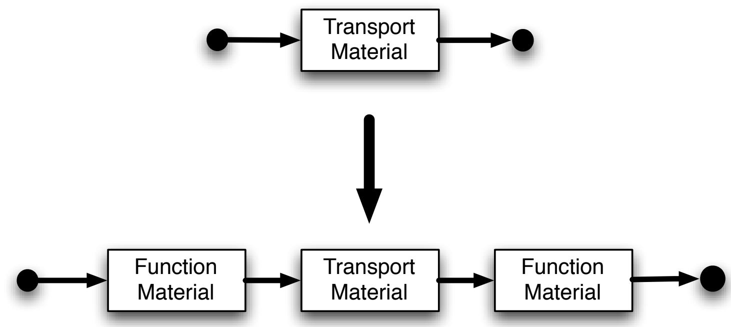


Figure B.4 Syntax number four, transporter.

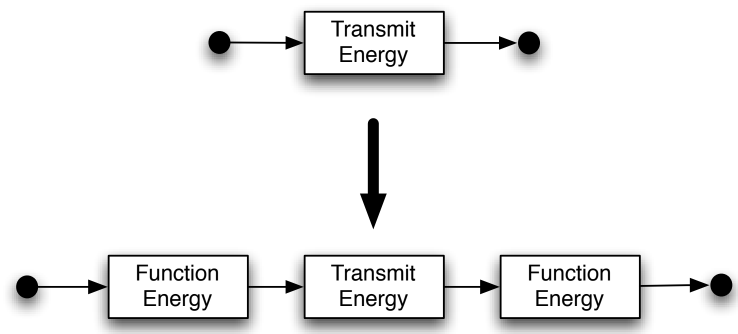


Figure B.5 Syntax number five, transmitter.

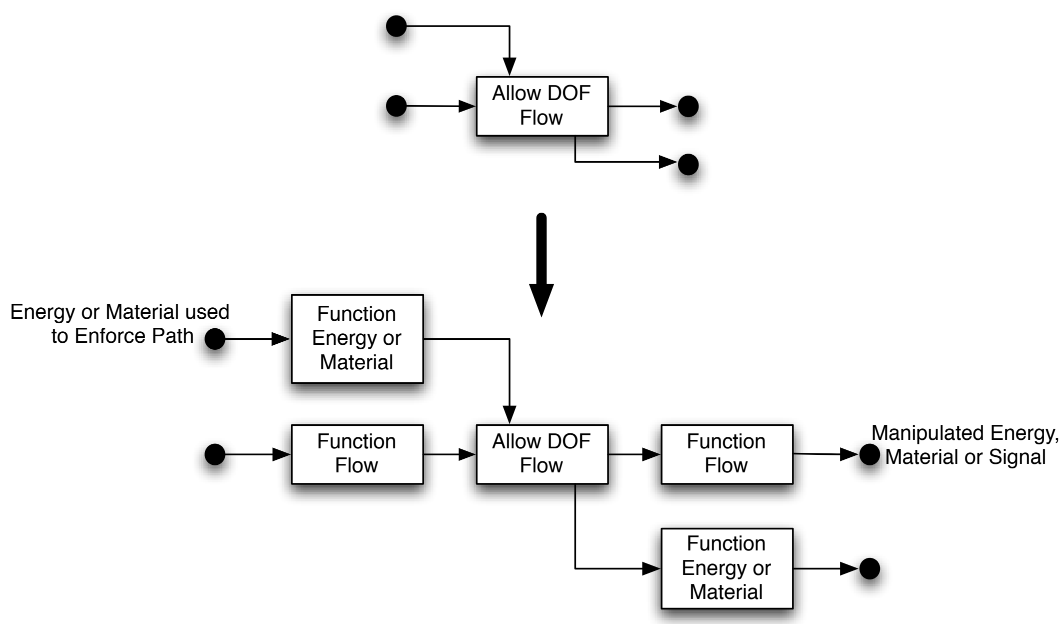


Figure B.6 Syntax number six, enforcer.

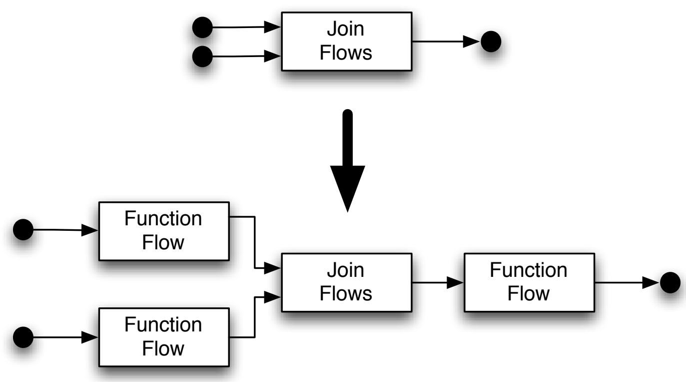


Figure B.7 Syntax number seven, joiner.

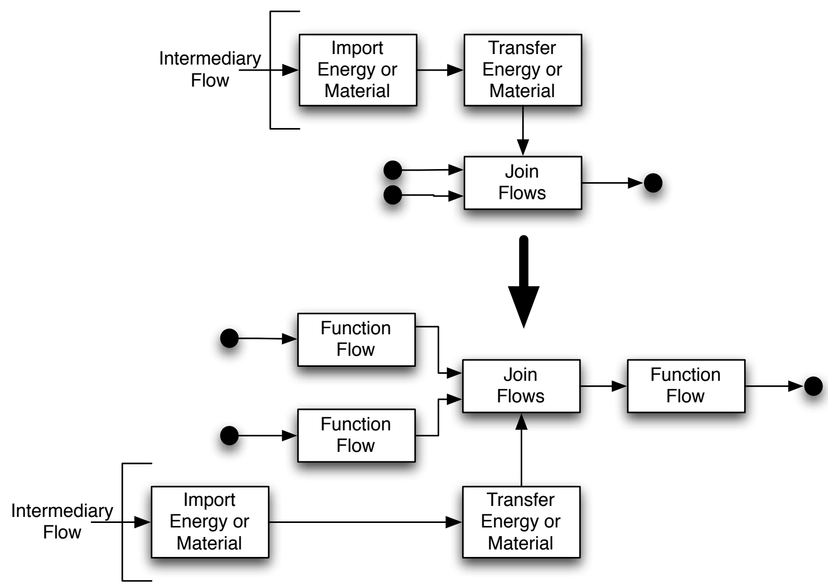


Figure B.8 Syntax number eight, linker.

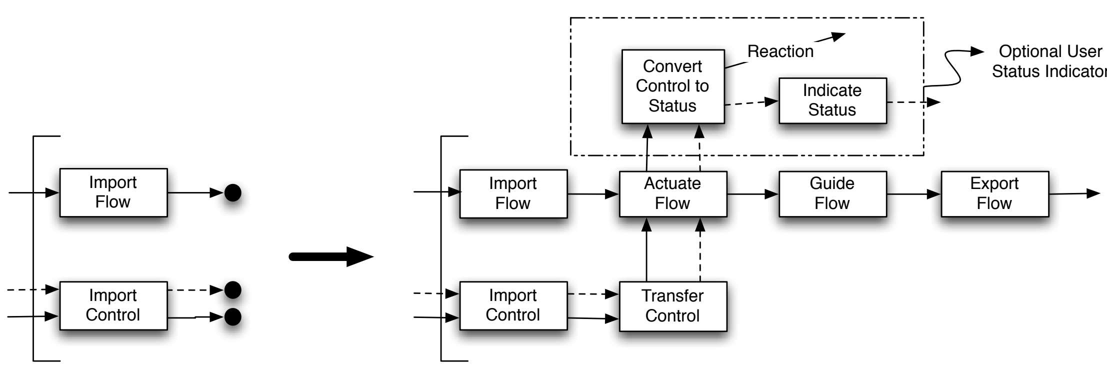


Figure B.9 Syntax number nine, actuator.

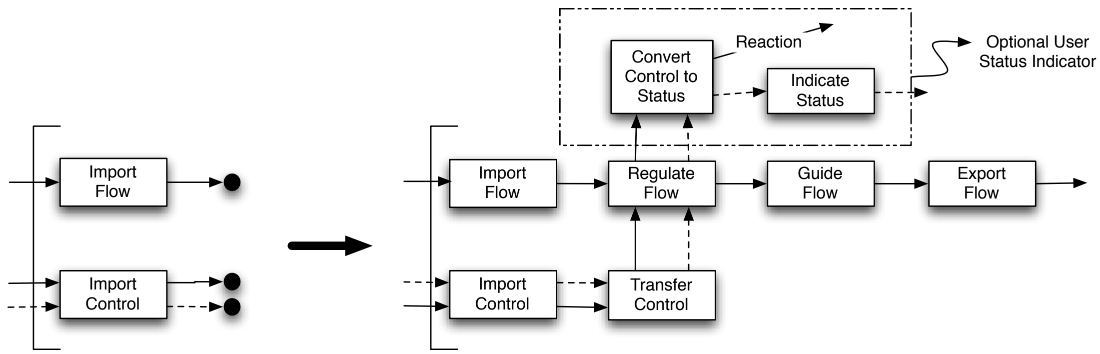


Figure B.10 Syntax number ten, regulator.

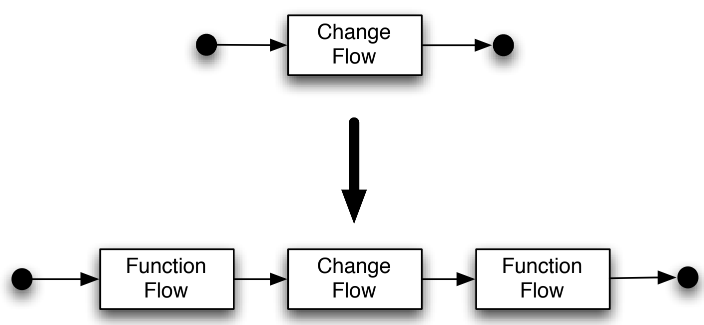


Figure B.11 Syntax number eleven, changer.

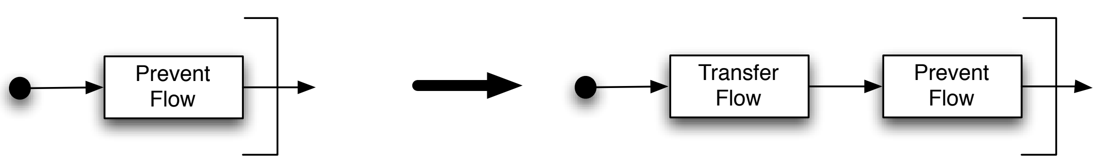


Figure B.12 Syntax number twelve, preventer.

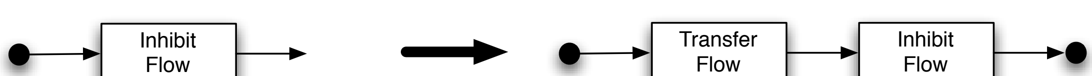


Figure B.13 Syntax number thirteen, inhibitor.

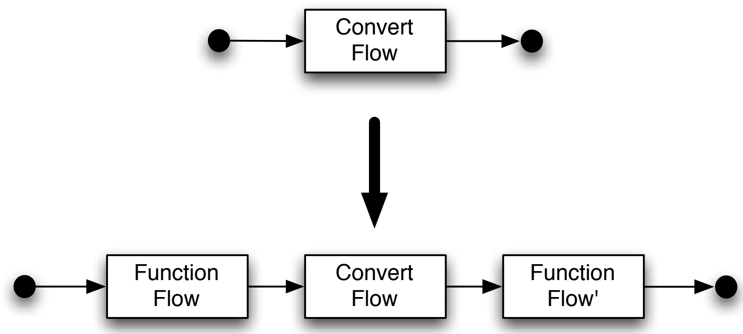


Figure B.14 Syntax number fourteen, converter.

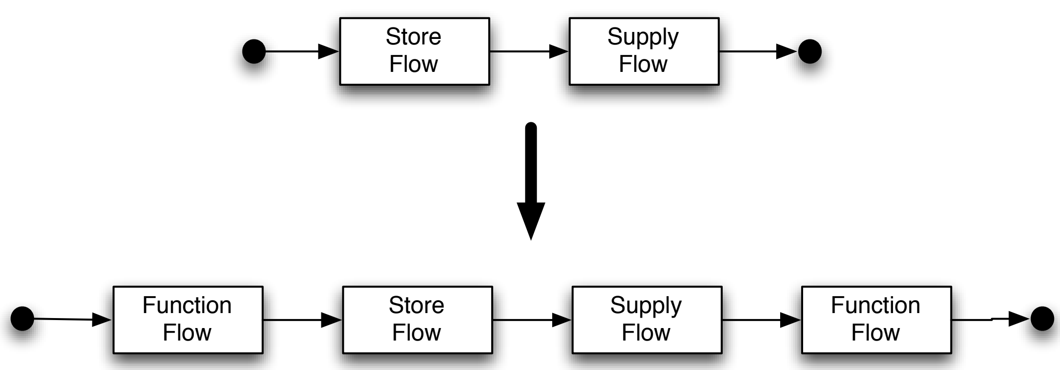


Figure B.15 Syntax number fifteen, provider.

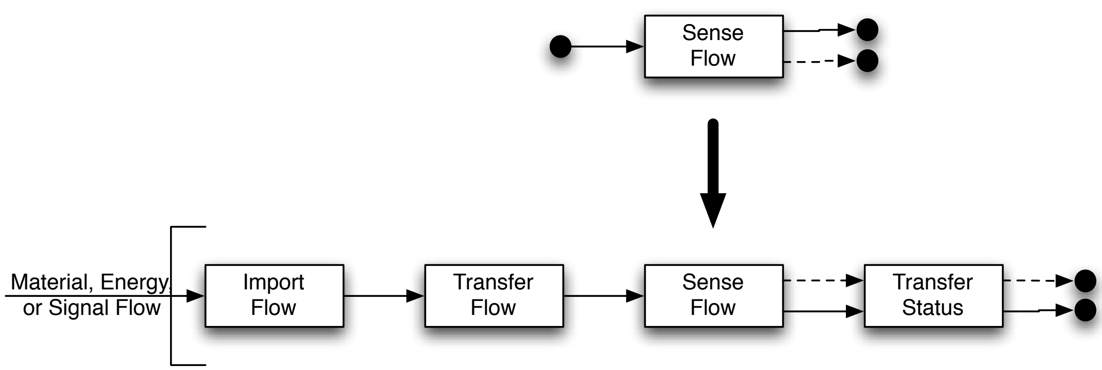


Figure B.16 Syntax number sixteen, sensor.

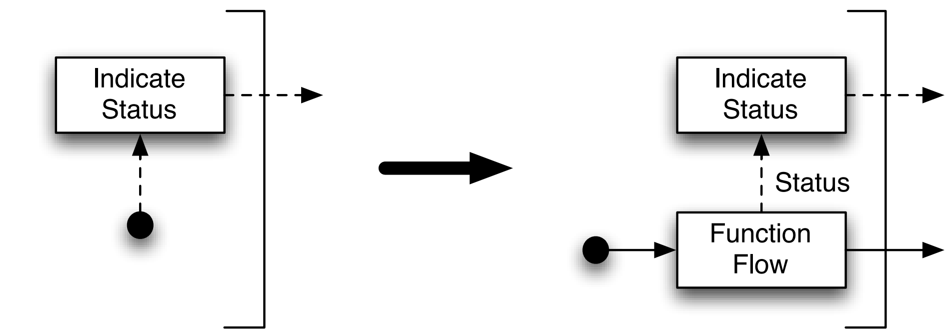


Figure B.17 Syntax number seventeen, indicator.

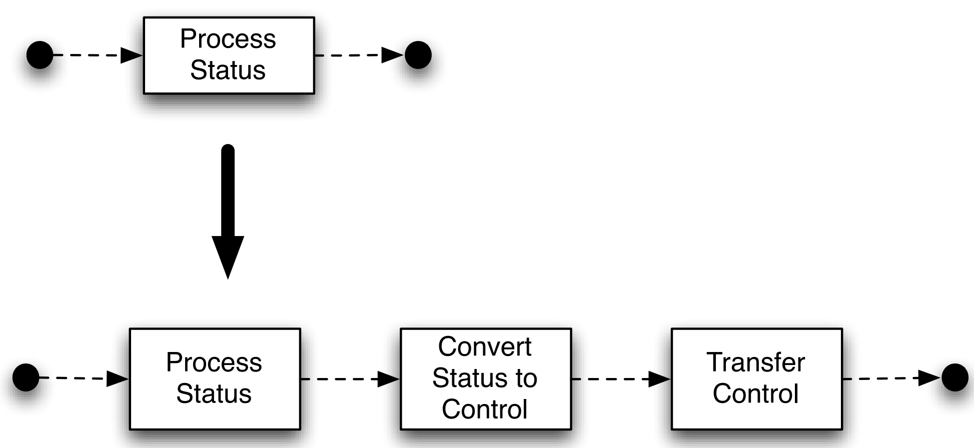


Figure B.18 Syntax number eighteen, processor.

C Full Page Functional and Configuration Models

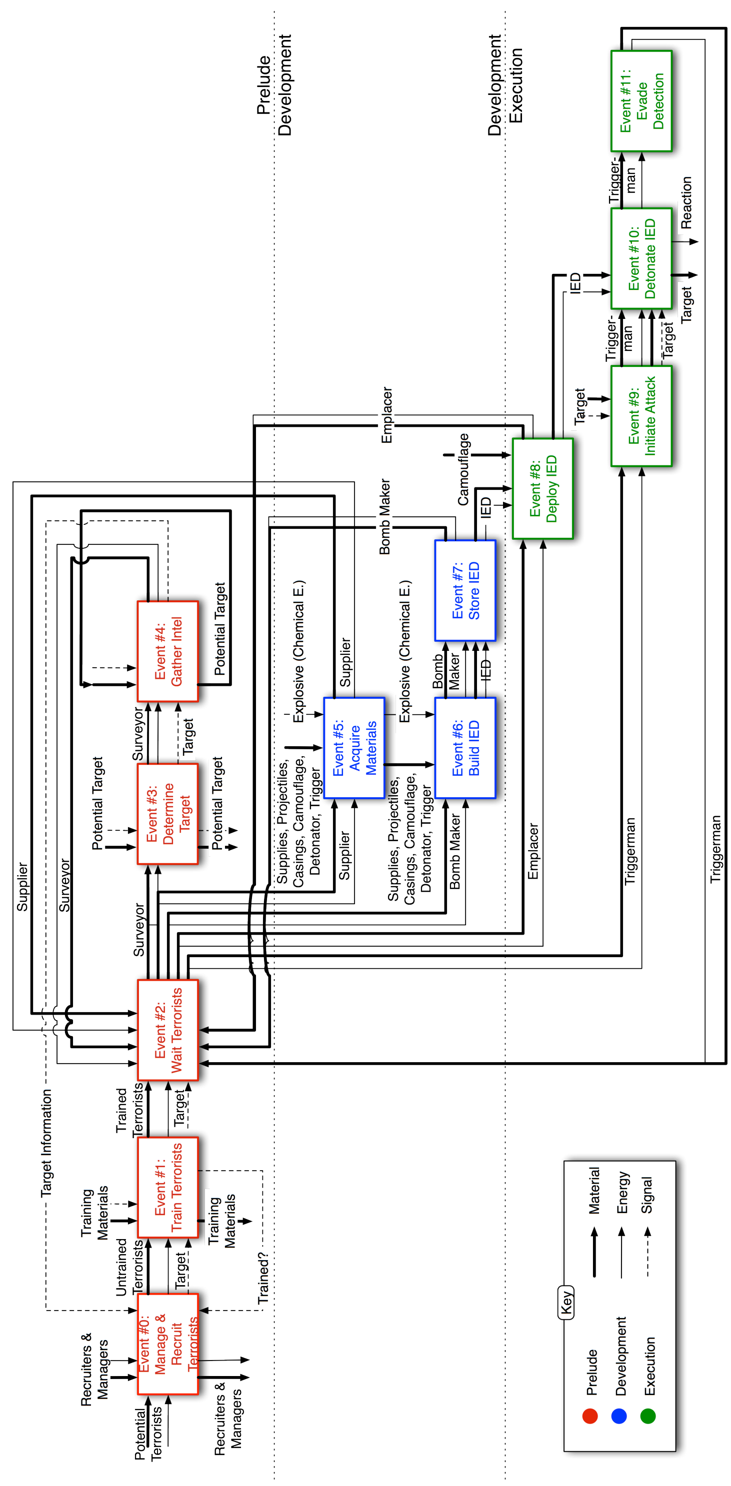


Figure C.1 Event model covering the three phases of an improvised explosive device incident.

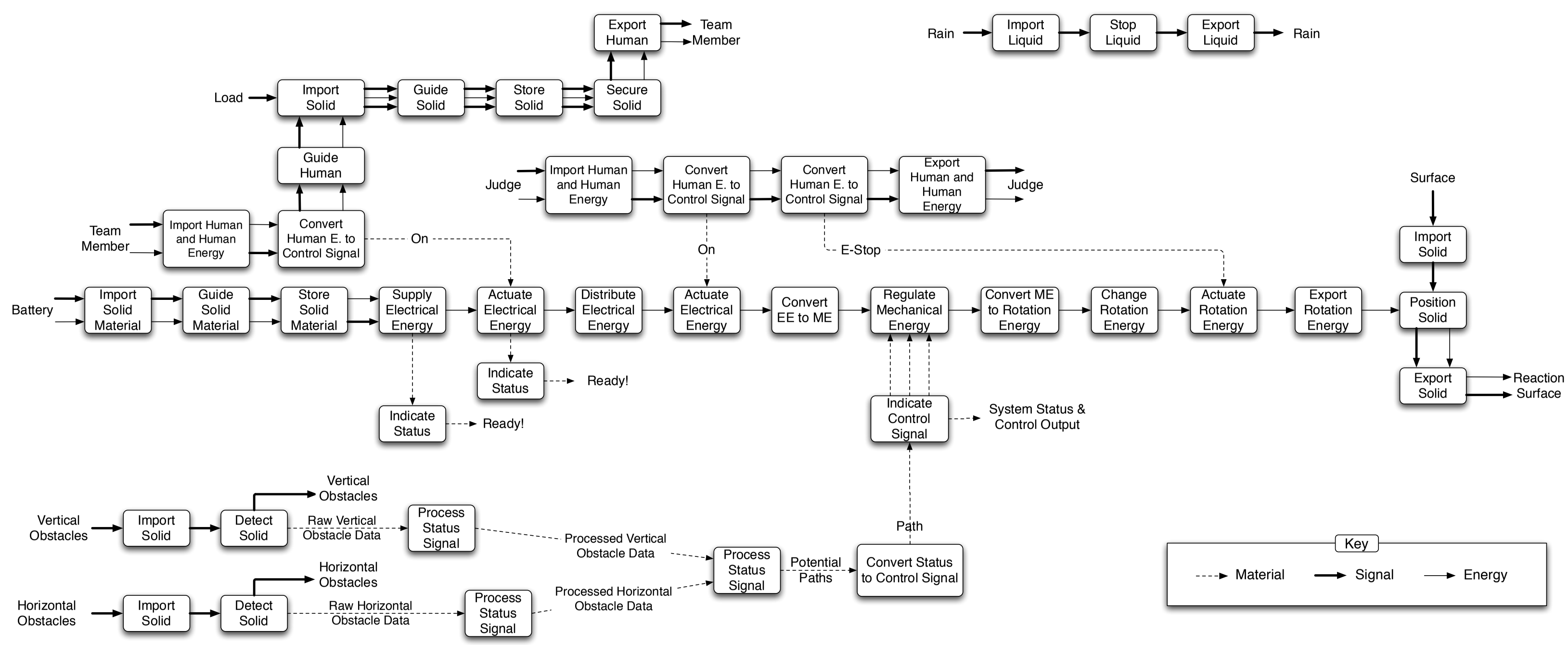


Figure C.2 The functional model of the conceptual design of the intelligent ground vehicle.

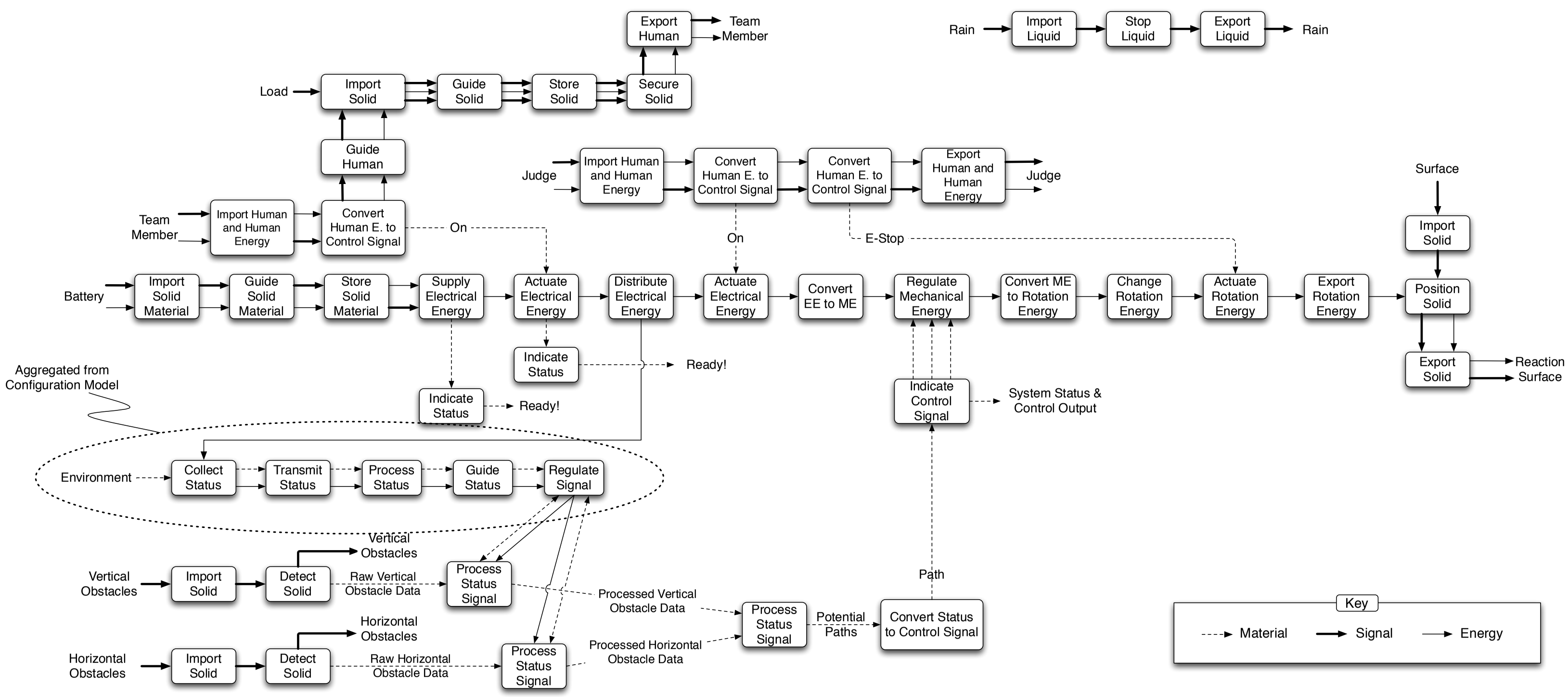


Figure C.3 The functional model of the intelligent ground vehicle with aggregated configuration chains.

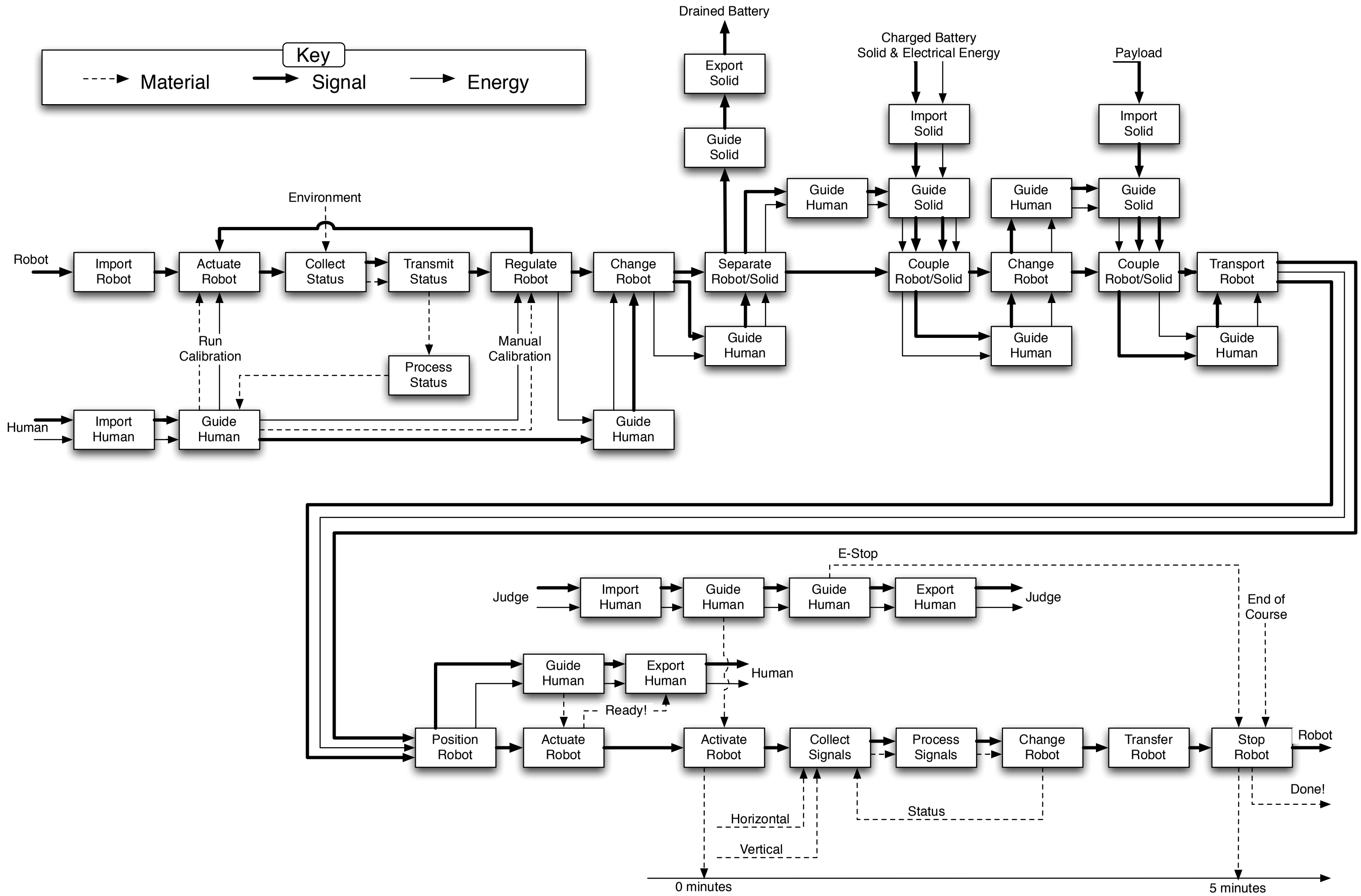


Figure C.4 Aggregated configuration model of the conceptual design of the intelligent ground vehicle.

VITA

Robert Lewis Nagel, son of Lewis and Linda Nagel, was born November of 1982 in Canton, Ohio. Robert attended grade school in Bedford, Texas and Mechanicsburg, Pennsylvania before graduating from Lake Central High School in St. John, Indiana in May 2001. Following high school, Robert attended Tri-State University (now known as Trine University) in Angola, Indiana. He received a Bachelor of Science degree in mechanical engineering in May 2005, graduating with High Honors and receiving both the Gold Key Scholastic Award and the Jannen Renaissance Scholar Award. Robert continued his education at the University of Missouri—Rolla (now known as Missouri University of Science and Technology), receiving a Master of Science in mechanical engineering in December 2006. Robert continued in the Doctoral Program at Missouri University of Science and Technology until the summer of 2009 when he transferred to Oregon State University in Corvallis, Oregon. He received his Doctor of Philosophy degree in mechanical engineering from Oregon State University in June 2010.