DESIGN FOR ADAPTABILITY

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

in the Department of Mechanical Engineering

University of Saskatchewan

Saskatoon

By

Mehdi Hashemian

PERMISSION TO USE

Head of the Department of Mechanical Engineering

University of Saskatchewan

57 Campus Drive

Saskatoon, Saskatchewan, Canada, S7N 5A9

ABSTRACT

Manufacturing globalization and sustainable development compel production enterprises to continuously seek improvements in their products' performance, customization, environmental friendliness, cost, and delivery time. The challenges of this competition cannot be completely addressed through improving production processes because some issues can only be solved through more innovative 'design'. This thesis investigates a new design paradigm called *Design for Adaptability* or *Adaptable Design (AD)* to address some of these challenges.

The purpose of AD is to extend the *utility* of designs and products. An adaptable 'design' allows manufacturers to quickly develop new and upgraded models or customized products through adapting existing designs with proven quality and costs. An adaptable 'product' can be utilized under varying service requirements thus prevents premature product replacement. *Design adaptability* and *product adaptability* provide economical and environmental benefits for AD.

To make a product adaptable, its adaptability must be built-in during the design stage. Methods of design for 'predetermined' adaptations are categorized as *Specific AD*; these methods design products for *versatility*, *upgrading*, *variety*, and *customization*. Several of these methods such as *modular/platform design* and *design for upgrading* have been studied for mechanical design. In the absence of predetermined adaptations, AD aims to increase the *general adaptability* of products. *General AD* involves fundamental research in design theory and methodology in order to develop practical design methods and guidelines. This thesis introduces several original concepts and proposes the *subordination of a system to a rational functional structure* as an approach for increasing general adaptability. Such a system would consist of a hierarchical assembly of autonomous functional modules, emulating the adaptable architecture of a 'rational functional structure'. Methods and guidelines are proposed for making the design of mechanical systems closer to this ideal architecture.

Accordingly, the thesis proposes a methodology for AD in which specific AD is performed first to take advantage of available 'forecast' information, and then general AD is performed in order to increase adaptability to 'unforeseen' changes. Also, a measure has been defined for the assessment of adaptability. The application of this methodology has been demonstrated through several conceptual design examples.

ACKNOWLEDGEMENTS

I express my most sincere gratitude to my supervisor Professor P. Gu for his guidance and inspiring ideas during my studies, also for his encouragement, advice, and support after I returned to the PhD program. This thesis would not be possible without him. I am grateful to my co-supervisor Professor G. Watson and to the members of the advisory committee at the University of Saskatchewan. I am also indebted to my friends and colleagues at the University of Calgary who shared their thoughts about engineering design research during many discussions and meetings, particularly to Mr. Neil Schemenauer who also programmed the software for the representation of functions in this thesis.

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the academic scholarship from the University of Saskatchewan are acknowledged and appreciated.

DEDICATION

Dedicated to my wife Ania

and to our son Hassan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

Increasing competition for better product functionality, quality, features, customization, environmental friendliness, lower cost and shorter delivery time presents unprecedented challenges for product manufacturing enterprises. These challenges cannot be completely addressed by utilizing advanced manufacturing technologies and optimizing production processes. Instead, companies are forced to improve the entire array of activities related to product development including marketing and problem identification, design, production, distribution, post-sale services, and environmental obligations such as recycling.

Of all the activities related to product development, design is considered to be the most important one. Various studies have concluded that a product's characteristics are primarily determined by its design, particularly by the decisions made during the early stages of the design process ([Ullman 1992], [Boothroyd 1994], [Kushnir 2003], [Simpson 1998], [Condoor 1999]). Therefore, much research in recent years has been dedicated to developing a fundamental understanding of the design process, improving design education, and devising tools and methods for assisting designers.

In this context, this thesis discusses *adaptable design* as a design paradigm for the economical success of the producer, the satisfaction of the customer, and the protection of the environment. Adaptable design, as the name suggests, aims at developing designs that are adaptable to various circumstances. Adaptability helps the producer reuse the existing design knowledge and manufacturing infrastructure, which is more cost effective than creating new designs and production processes. Adaptability allows the user to utilize the same product under varying circumstances, hence replacing several products with one. Adaptable design is also beneficial to the environment because it reduces the total production volume and instead develops products that yield more service than their conventional counterparts.

In this introductory chapter, the first section is dedicated to providing some background information. This section first discusses an important premise of this research, which is the treatment of 'adaptability' as a design characteristic. This leads to 'design for adaptability' as a new design paradigm. Next, this section discusses the current state of research and identifies 'general adaptability' as an original research topic. Section 1.2 briefly discusses the principle of segmentation (modularization) which is fundamental to the approach of this thesis. Section 1.3 is dedicated to clarifying the distinction between adaptable design and modular design. Section 1.4 provides an overview of the thesis in the logical sequence of main ideas. This is followed by the thesis organization and list of terms at the end of this chapter.

1.1. Background

Adaptability as a Design Characteristic

There are practical and economical benefits in the ability to adapt a product to different service conditions. For example, it would be useful if we could adapt a car to varying driving needs, adapt a CNC lathe to better technologies that become available, or adapt a single good design to different sets of requirements and thus produce several different products. Adaptation becomes particularly beneficial when a product would be put out of service while it is in good working condition. Such premature retirement of products might be caused by changes in the needs or expectations of the user, by changes in operational conditions or government regulations, and increasingly in the modern engineering market, by the technological obsolescence of components. In such cases, adaptation creates new service life for products which otherwise would be disposed of.

Despite such obvious advantages of adapting products, adaptation is not always practically possible. Some adaptations can be performed at reasonable cost and effort, for example the adaptation of a personal computer to the new technologies such as faster CPUs and larger memories. Some other adaptations, on the other hand, are too expensive or difficult to be practical, for instance the adaptation of a car for a different number of seats or for a different location of the driver in the vehicle.

The difficulty of adapting a product to a new set of service conditions depends on the differences between the new service and the original service, as well as on certain attributes of the product that determine how easily the product can be altered from its current state to the required new state. Examples of these attributes include the way the product is divided into subsystems, the way its various subsystems are connected, and the possibility of altering the configurations and functions of various components. The collective effect of these attributes can be viewed as a product's ability to be adapted to new service conditions. This characteristic can be called the product's "*adaptability*" [Gu 2002].

Thus, adaptability can be treated as a characteristic similar to *manufacturability*, *recyclability*, or *upgradeability*. Similar to these characteristics, the adaptability of a product depends on many specifications and attributes of the product's subsystems and components, hence is easy to describe and understand but is difficult to quantify. Also similar to these characteristics, the adaptability of a product is primarily determined by its 'design'.

Design for Adaptability

A design paradigm is a theoretical framework for designing; it may include rules and generalized methods, guidelines, specific procedures, software tools, etc. Examples of engineering design paradigms include *concurrent engineering*, *systematic design*, and *decision-based design* [Pratt 1993]. In the past few decades, several paradigms have been developed for the purpose of improving specific characteristics of products during the design process. These are known as *Design for X (DFX)* (e.g. "Design For Assembly", [Boothroyd 1983]). A DFX paradigm helps designers develop products which are likely to perform better with respect to characteristic X. For example, *design for manufacturing* and *design for recycling* are established paradigms that help designers develop products with greater manufacturability and recyclability.

The goal of this thesis is to contribute to the development of a new DFX paradigm, one which aims at developing products with greater 'adaptability'. This paradigm can be called *design for adaptability*, or *adaptable design* (AD). While a conventional mechanical product is designed to serve in its normal operational mode, an adaptable product is designed to be able to change its operational mode in some circumstances.

Current State of Research

Design for adaptability is not an established paradigm in mechanical engineering design;

therefore there is a shortage of direct literature on the subject. There are, however, several design methods whose objectives are to increase various types of adaptabilities in mechanical designs. These methods are presented under different titles in the literature.

One way of locating research pertinent to AD is to categorize various scenarios of adaptations, then search for the existing design methods which aim at facilitating these scenarios. This approach, to be discussed in detail in Chapter 3, results in the identification of four objectives among the design methods which are related to AD: *upgrading*, *variety*, *versatility*, and *customization*. Here, "upgrading" refers to adaptations that occur over the course of time; "variety" and "versatility" refer to the adaptability of designs and products respectively; and "customization" is a general term used in the literature to refer to all these scenarios. By this definition of terms, the existing design methods related to AD can be categorized under the following four paradigms: *design for upgrading*, *design for variety*, *design for versatility*, and *design for customization*.

For instance, methods of *design for upgrading* are those which aim at facilitating future adaptations of artifacts; these methods postpone the retirement of products and extend their service life. Often these methods focus on the technological obsolescence of components. In such cases the process of upgrading typically involves the replacement of expired parts. The common method for facilitating this process is to design the rapidly-expiring subsystems as replaceable modules. A very successful implementation of this method can be observed in the design of personal computers. Soon after the initial introduction of PCs to market, it became evident that their premature retirement was an issue. A typical PC has a relatively long product life because it does not undergo much wear and damage, yet it has a short service life due to the rapid technological obsolescence of its components. Therefore, PCs would be disposed of while in good working condition. The utilization of a modular architecture in the design of PCs avoids this problem. In this architecture, rapidly expiring parts such as the CPU or the memory card are designed as separate modules and can be easily replaced with newer ones.

As the PC example shows, the methods of design for upgrading typically assume that future upgrades are known at the time of design, so that the subsystems which are bound to be replaced in the future can be designed as detachable modules. The review of the other design methods related to AD reveals that they also assume that future adaptations are known in advance thus can be "designed-in" at the beginning of product planning. For example in 'design for variety' a common method is the development of shared *platforms*, based on which a family of products can be created through the addition of *differentiating modules*. In this procedure, it is assumed that product variations are foreseen at the time of design, therefore their commonalities can be developed as shared platforms.

Specific and General Adaptabilities

The above discussion presented an important observation that the existing design methods, though diverse in their objectives and techniques, have an element in common which is the assumption of forecast information during designing a product for future adaptations. Since any of the current methods targets a specific set of adaptability objectives from the outset, this thesis uses the term "*specific adaptability*" as an umbrella term to refer to the aim of the existing design methods. The methods of design for specific adaptabilities are very helpful, but generally they are only applicable to their foreseen adaptations. There are certain design characteristics that make one product generally more adaptable, even to unforeseen changes, than another product with similar functions. We use the term "*general adaptability*" to refer to these characteristics.

Currently, a formal approach towards designing products for general adaptability is not available in the mechanical engineering design literature. This is primarily due to the inherent properties of the mechanical design process, to be discussed in Chapter 4. As a result of these properties a typical mechanical system is designed for a specific operational mode. In such a system the overall functions are achieved through the interactions among many subsystems and components which are often useful only in their exact configuration. Therefore, the structural or functional alterations which are necessary for an adaptation task are typically very difficult to make.

In design methods for specific adaptabilities, the overall strategy is to provide for the features which are needed for a 'predetermined' set of adaptations. A design method for general adaptability naturally requires a different strategy because in the absence of forecast information no particular adaptations can be targeted during the design process. In order to develop a design method for general adaptability, this thesis first takes a theoretical approach and presents an ideal architecture for general adaptability. Then it shows how the new technologies can be utilized to overcome the inherent difficulties of mechanical design and develop mechanical systems which emulate this ideal adaptable architecture as closely as possible.

1.2. Segmentation (Modularization)

There are several techniques for enhancing a design with respect to the specific adaptabilities discussed above. These techniques include *modular design*, *product family development*, and *platform design*. It can be observed that the underlying principle in these techniques is the *segmentation* of a product. In a segmented (modular) product the alterations made in one place are likely to be confined within one or a few segments; whereas in a product with a more integral architecture the alterations made in one place are likely to propagate to the rest of the product. Therefore, a product with a segmented architecture is generally easier to modify, hence has greater adaptability.

In the segmentation methods for specific adaptabilities, where future changes are known in advance, the main task is to find a segmentation scenario that yields the best results with respect to the target objectives. For example, in *design for variety* the segmentation criteria are commonality and differentiation. That is, those subsystems which are shared among a family of products are grouped together as a common platform, and the differentiating features are developed as add-on modules. As another example, in *design for upgrading* the segmentation criterion might be obsolescence. That is, the rapidly-expiring subsystems are developed as replaceable modules.

Given the effectiveness of segmentation in achieving specific adaptabilities, this thesis explores the use of the same principle for achieving general adaptability. Therefore, the main task is to find a segmentation scenario that yields greater general adaptability in a design. For this purpose, this thesis suggests the use of *functions* as the criterion for the segmentation of a design. That is, the physical subsystems of a product are divided in such a way that every subsystem performs a useful function.

Since the design process begins from the functional domain and proceeds to the physical domain, the *function-based segmentation* of a design can be viewed as the *subordination of the physical structure of a product to its functional structure*. Function-based segmentation is the main method of this thesis towards achieving general adaptability. This method, along with guidelines for implementing it in the design of mechanical systems, will be discussed in Chapter 4.

1.3. Modular Design and Adaptable Design

In the literature, the term *modular design* is often used in its broad sense to refer to all methods of segmentation. For example, platform design might be considered a special case of modular design in which common subsystems are developed as a shared module. Modular design, however, is a different concept from adaptable design despite the fact that modularization is also the main method for increasing adaptability.

Segmentation of a product in modular design might be performed for various objectives, including those related to adaptability such as upgrading and those unrelated to adaptability such as material recycling. Adaptable design, on the other hand, may or may not use the method of segmentation but its objective is invariably related to adaptability. The following figure illustrates the relation between these two concepts.

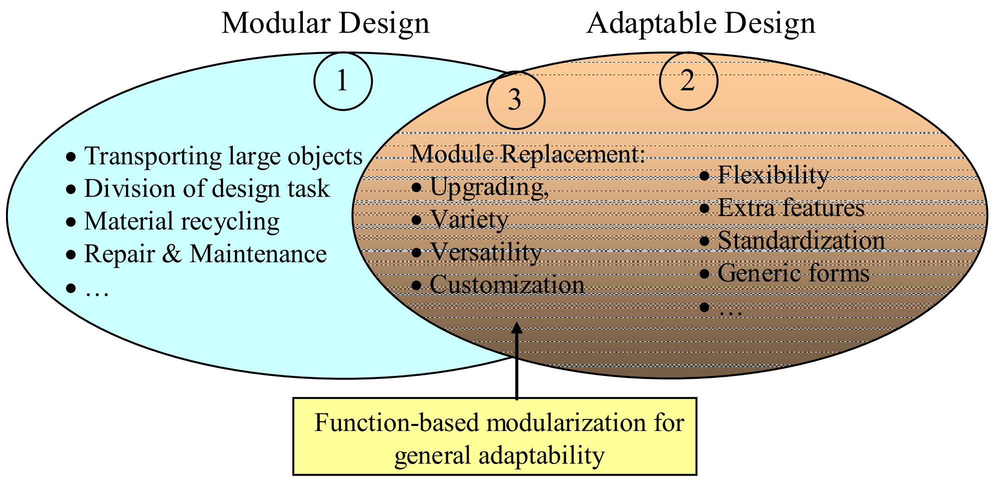


Figure 1. 1: Modular Design and Adaptable Design.

Area 1 on the left includes those objectives of modular design which are not related to adaptability. These objectives will be discussed in Chapter 2. Area 2 includes methods of increasing adaptability which are not related to modularization. One such method is to utilize flexible elements in a design, where flexibility refers to the ability of a system to be reconfigured without breaking it into its constituent subsystems. Examples of such integral flexibility can be seen in natural systems such as human skin, and in man-made objects such as hydraulic hoses, manufacturing adjustments, and flexible shafts. Another method is the provision of extra features and functionalities in a design for possible future needs. For example, an output utility shaft is provided in the design of most farm tractors; thus making the tractor adaptable to various functions. The applicability of this method is often limited by the higher cost of excess functions. Other methods in this area include increasing the compatibility among subsystems through the standardization of systems and their interfaces, and using more regular surfaces and generic forms which facilitate future alterations and amendments.

Area 3 is the overlap between the two concepts. Methods in this area utilize the principle of segmentation to localize the structural alterations made for the purpose of adapting the product. This area includes the conventional methods of modularization for upgrading, versatility, variety, and customization, as well as the function-based segmentation method proposed in this thesis for achieving general adaptability.

1.4. Thesis Overview

Definitions

Given the fact that AD is not an established research topic in mechanical engineering design, this thesis inevitably presents a few definitions and categorizations. This task begins by identifying the '*extension of utility'* as the purpose of adapting an artifact. This view reveals a logical relationship between the scarcity of resources and the need for adaptability. This view also helps to identify various categories of adaptations by considering different modes for extending utility, as mentioned in the background section. These topics will be discussed in Chapter 3.

The extension of utility might also be the purpose of many tasks which are not considered adaptation. Therefore, the next step is the adoption of limits for the scope of tasks related to AD. Three such limits are proposed in this thesis. First, adaptation is the extension of usage into 'different' operational modes. This limit excludes from AD the methods of prolonging the normal service life of products, such as the methods of design for durability, repair, and maintenance. Second, adaptation is a process in which components are utilized as they are. This limit excludes from AD the methods of recovering artifacts in other forms, such as material recycling. Third, an artifact is considered 'adapted' if a considerable portion of it remains together in the new operational mode. This limit excludes from AD the methods of design for part reuse. It can be seen that these limits are relatively subjective, for example the distinction between part reuse and adaptation might be unclear. However, these limits resolve some inconsistencies that exist in the usage of the term 'adaptability' in the literature, and help in better determining the scope of AD in this research.

After these definitions are established, the next step is to seek design methods for increasing various types of adaptabilities in mechanical systems. Current methods are identified and classified under four categories discussed in the background section. These are design methods for increasing the upgradeability, customizability, versatility, and variety of mechanical systems. In any of these methods, a set of future adaptations is targeted from the outset, and then the product is designed in such a way to facilitate these foreseen adaptations.

It could be observed that there are no formal design methods for designing mechanical systems for general adaptability, that is, without *a priori* aiming at a specific set of adaptations. Therefore, AD could be divided into *specific AD* and *general AD*. Specific AD refers the design methods discussed above, and general AD refers to a method for designing adaptability into mechanical systems without targeting particular adaptations.

General AD

As discussed in previous sections, 'segmentation' is the principal method for achieving adaptable architectures. In the methods of specific AD, the segmentation criteria depend on what adaptations are targeted during the design process. For example 'average service life' and 'commonality' are the segmentation criteria in design for upgrading and in platform design. In general AD, however, different criteria are needed because adaptations are not predetermined. This thesis proposes the use ‘functions’ as the segmentation criterion. That is, the subsystems of a product are divided in such a way that each subsystem independently provides a useful function.

The reason for choosing functions as the segmentation criterion is that a carefully constructed functional structure can be very adaptable. Chapter 4 presents a model of design process for this purpose. This scale-independent model follows a decomposition process similar to the one which has been described by Suh in Axiomatic Design [Suh 1990]. In this process, no function is decomposed unless a solution for it is conceived first, and any set of sub-functions must be both necessary and sufficient for describing the requirements of their parent function. Beginning from the initial problem, recursive decomposition of functions results in a hierarchical structure. In this hierarchy, the relation between a function and its subordinates is a *causal* relation. That is, every function in the hierarchy exists only to assure the proper functioning of the adopted solution for its parent function. In an ideal functional structure, there are no causal relations between functions except for those between a parent function and its sub-functions. Therefore, an ideal structure demonstrates two properties when a function is removed from it. First, this function’s sub-functions and their subordinates become unnecessary and can be eliminated as well. Second, there is no need to eliminate anything else from the structure. Due to these properties, when such a functional structure is modified in some location, these modifications do not propagate to the rest of the structure. This makes such a structure ideal for adaptation tasks.

Therefore, the approach of this thesis is to imitate the properties of an ideal functional structure discussed above in the actual design of a product. This ideal functional structure is called the '*rational functional structure*' in this thesis because the decomposition process that generates this structure represents the design rationale. Thus the proposed method can be equivalently described as *'function-based segmentation'* or *'the subordination of the physical structure to a rational functional structure'*. In this method, first the '*physical functions*' that are expected from the final product are distinguished from non-physical requirements. Several sets of physical functions can describe the purpose of the product; the designer should choose the set in which functions are as useful and recurring as possible. Then, for each function a separate solution is devised which is self-contained and autonomous in delivering that function. Such a module should not depend on a particular configuration for its operation, and should be useable anywhere as long is its proper inputs and outputs are provided. If a module requires further decomposition and problem solving, the above two steps are repeated when possible, attempting to develop the subsystems of this module also as autonomous and functional modules. The result will be a hierarchy of physical modules that corresponds to the hierarchy of functions in the rational functional structure.

Frame-and-Function Architecture

The functional modules developed in the above procedure, along with product-specific components if applicable, have to be assembled into the overall product. This assembly must fulfill the embodiment requirements of the design problem. The overall shape of the mechanical object can be determined by the spatial positioning of assembled components, and in some cases, by an actual skeletal structure. If we use the term 'frame' for the spatial provisions of the overall embodiment of the product and the term ‘function’ for the functional modules, then the architecture of a product designed by the above method can be called *‘frame and function’*. This architecture is an alternative to the *platform architecture* which is commonly used in design for variety and customization. This will be discussed in detail in Chapter 4.

Guidelines

The development of an artifact through the assembly of functional modules can be seen in a business model called the *virtual enterprise* and in object-oriented software programming. The achievement of this architecture in mechanical systems, however, might be very difficult or even impossible. Chapter 4 presents several guidelines which help with implementing this method in the design of mechanical systems. Examples of these guidelines include:

- Subsystems should be functionally autonomous and their functions should be meaningful and recurring.
- The design should begin from the components that interact with the environment and then proceed to develop the necessary internal mechanisms.
- If possible, functions should be achieved by software not by hardware.
- Physical dependencies among various assemblies should be minimized (e.g. by using flexible interfaces and manufacturing adjustments).
- Standard components and generic forms should replace product-specific designs when possible.
- Extra features which help with future adaptations and do not add to the cost should be considered.

The AD Methodology

The overall methodology of 'design for adaptability' combines specific AD and general AD. The procedure first identifies the functional requirements of the original design, as well as the requirements that are related to future adaptations if such forecast information is available. Next, a functional structure is developed together with the conceptual design of the overall product. The product is designed in such a way to provide for both the functional requirements of the design and the requirements of targeted adaptabilities. This step assures that the forecast information is utilized and that specific adaptability is given a higher priority than general adaptability. Then, the product is further developed using the general guidelines of AD.

Measure of Adaptability

Function-based division of the subsystems of a design might compromise other criteria such as aesthetics. However, as the increasing scarcity of resources raises the tendency towards a service-based market, the utility of a mechanical system, which is to perform a physical function, becomes more important than other criteria. Further, the increasing prices of natural resources and the increasing affordability of advanced technologies will justify AD for a greater range of products in the future. These arguments will be discussed in Chapter 3. At any case, the thesis proposes a measure for adaptability which can be used in trade-off decisions.

The adaptability of an artifact for a particular adaptation task is calculated based on the savings that are achieved through adapting an existing product, as opposed to obtaining a new one. The calculation of saving is based on '*information content*', which in this thesis is defined as an indication of '*total costs*'. Since monetary means are the closest available tool for representing the total costs of activities, the measure of adaptability in our case studies is based on financial savings. These will be discussed in Chapter 4.

1.5. Thesis Objectives

The objectives of this research can be categorized under three main goals:

Establishing Research Framework

The first aim of this research is to establish a framework with definitions and categorizations which are needed for exploring "design for adaptability" as a new design paradigm.

Fundamental and Theoretical Research

The second objective of this thesis is to explore the fundamental principles of design and establish a theoretical basis for AD. This involves developing a model of the design process and a method for General AD. This leads to the development of a methodology which includes Specific AD, General AD, and a method for assessing adaptability.

Methods and Guidelines

The third goal of this research is to develop practical methods and guidelines for the application of the proposed methodology in mechanical engineering design problems.

1.6. Organization of This Thesis

Chapter 2 reviews the literature on engineering design research. Chapter 3 discusses the objectives of adaptability in engineering systems and the benefits of AD from the user, producer, and environmental perspectives. Chapter 4 presents the central arguments and methods of this thesis; it includes a method for measuring adaptability and the methods and guidelines for adaptable design. This chapter is largely self-sufficient as some concepts that have been discussed in Chapter 3 are also briefly repeated in Chapter 4 when required. Chapter 5 provides several examples for both specific AD and general AD. It is followed by a summary of contributions and conclusions in Chapter 6.

1.7. Terms and Definitions

The following is a list of terms which are used for specific meanings in this thesis.

- **AD**: Adaptable Design, which refers to Design for Adaptability.
- **Adaptability**: the ability of a product to adapt to varying service requirements.
- **Artifact**: the means of accomplishing a human purpose. In engineering design an artifact is a physical device created for the purpose of performing a function. Although in some cases the physical object itself is the purpose of creating it; in a service-based market only the function is the purpose of creating an artifact.
- **Decomposition**: the process of introducing new requirements in a design problem in order to assure the proper functioning of adopted solutions for the functional requirements of the problem.
- **Design**: as a noun, *design* refers to a blueprint or a plan or a recipe for accomplishing goals. In engineering, design is the set of instructions for the manufacturing of a physical object.
- **Environment**: a system that includes what is relevant to a goal, hence relevant to the artifact that achieves that goal.
- **Frame**: the abstract or actual skeleton that determines the overall embodiment.

- **Frame-and-Function Architecture**: a temporary assembly of functional modules on a spatial frame for a temporary service.
- **Function**: the actual physical effect of a component on material, energy, and signal. The function of an engineering system is assumed to be its *raison d'etre*, thus function is the goal of designing, producing, and using a product.
- **Functional Structure**: the hierarchy of functions, with initial function requirements in the apex and the decomposed functions in mid and end levels. This hierarchy is also called the '*rational functional structure*' because it is generated through the processes of problem solving, decision making, and decomposition. This definition is consistent with the definition provided in *Axiomatic Design*. It is, however, in some cases not consistent with the more popular definition provided in *Systematic Design*. Systematic Design defines the functional structure as a description of the operation of an existing device, not as the representation of the device's purpose or the design rationale.
- **Information Content (IC), Total Costs**: the IC of a design is the total amount of resources which are needed in order to achieve goals according to that design. This includes human and intellectual resources, technological expertise and infrastructure, natural resources, space, materials, energies, etc. The IC of a design is also equivalently called its Total Costs in this thesis.
- **Information Processing** (IP): the work or effort spent for performing a task. Any work is equivalently treated as 'information processing' or 'allocation of resources' in this thesis.

- **Information Processing Capacity** (IPC): a limited resource that represents the amount of work or IP which is available for performing a task.
- **Objective**, **Goal, Purpose**: a preferred state that would not automatically occur without our intentional intervention, that is, without human design.
- **Operational Mode**: the operational specifications of a product including working conditions, functions, performance characteristics, user interface, physical attributes, etc.
- **Physical Structure**: the hierarchy of physical assemblies in a product.
- **State**: the attributes of the environment including its dynamic trends.
- **Task**: the process of accomplishing a set of goals. This definition is utilized when the concept of information content in this thesis is introduced.
- **Utility, Service**: the purpose of a product, which is to perform a function.

Chapter 2: Literature Review

In this chapter, Section 1 reviews some of the theoretical research in engineering design and Section 2 reviews the research relevant to product configuration design. Research in this category is related to *design for specific adaptability*, as discussed in Chapter 1. A brief discussion on the current state of research related to AD is provided in Section 2.3.

2.1. Theoretical Engineering Design Research

In recent years, both academia and industry have realized the growing importance of structured, scientific, and industrially tested theories and methods for design. However, theoretical research towards establishing the science of design has not produced enough practical results [Tate 1995]. Design research is expected to yield practical benefits whereas research on natural sciences is not necessarily relevant to practice ([Reich 1992], [Broadbent 1981], [Cross 1980]). In discussing the current research towards a scientific theory of engineering design, Dixon states that a scientific theory is the ultimate goal of design research, and that the design research is in a pre-theory stage [Dixon 1988]. His view is that cognitive studies involve far too many ill-defined variables to support a theory and that prescriptive models are premature until they can be based on a validated theory. He also discusses how pre-theory research in engineering design can advance towards that goal. Finger and Dixon state:

"*One school of thought believes that it is necessary to develop theories of design… Another school believes that the matter of design theory is an irrelevant distraction… Some consider that the best path is to concentrate on those portions of the design process for which we already have useful theories and tools, such as Decision Theory and Optimization…*" [Finger 1989].

Several researchers have contributed to the establishment of scientific principles for various aspects of the design process. Examples include the works of Yoshikawa and Tomiyama on a general design theory, the mathematical representation and a general design governing equation developed by Gu and Zeng, and Suh's axiomatic design theory ([Yoshikawa 1981], [Tomiyama 1996], [Suh 1990], [Zeng 1999-a], [Zeng 1999-b]). Various other approaches are also of a theoretical nature. Examples include models of the creative process of design ([Altshuller 1984], [Dorst 2001]); cognitive models ([Kolodner 1983], [Ullman 1987], [Condoor 1992]); Robust Design, House of Quality, and other techniques of clarifying design objectives and assessing their attainment ([Taguchi 1987], [Clausing 1994], [Ramaswamy 1992], [Belhe 1996], [Chen 2002]); representation of technical functions ([Cross 1980], [Hashemian 1997-a]); and systematic design ([Pahl 1988], [Hubka 1996]). This section provides a review of some of these theoretical approaches towards engineering design.

2.1.1. Descriptive (Cognitive) Models

Research on the nature of the engineering design process may result in the development of a *descriptive model* of design. The basic task in developing descriptive models is to study how humans create designs in order to determine what processes, strategies, and problem solving methods designers use. One way to develop descriptive models is to conduct *protocol studies* [McNeill 1998]. The challenge with such protocol studies is that much of the design process happens in the mind of the designer and thus the documentation does not reveal the entire thought process. Further, the process of documentation may interfere with the designer's work during the design process.

Despite these shortcomings, protocol studies have revealed some facts about the nature of design [Finger 1989]. Some of these studies suggested that designers need the help of different tools depending on their experience with the design task and the stage of development during the design process. For example, information retrieval tools are needed when prior experiences with similar designs exist, as in case-based design [Kolodner 1993]. Another study by Ullman revealed that designers tend to pursue a single design concept, and that they will patch and repair their original idea rather than generate new alternatives [Ullman 1987]. If prior experience exists, designers reuse familiar solutions and will not explore alternatives or innovative ideas unless their new design fails badly and cannot be salvaged. These observations reveal the impact of bias or the so-called *psychological inertia* on the designer's work. Another interesting observation is that design is a fractal-like process in which the stages of design repeat continuously at different times and at different levels of detail [Ostrosi 2003].

Cognitive research has also revealed that during the design process, designers use general prototypes and adjust them to the particular demands of a given problem. Maher created a system that represents design knowledge as prototypes [Maher 1987]. The scope of this method of representing design knowledge, however, is limited to problems in which synthesis consists of only a morphological combination [Hashemian 1997-b]. In another study, Maher and Tang presented a protocol study of human designers looking for evidence of co-evolution in design, that is, the simultaneous evolution of both the design problem and the design solution during the design process [Maher 2003]. Co-evolutionary design is developed as a cognitive model of design in which designers iteratively search for a design solution and make revisions to the problem specification. They discuss the similarities and differences between this cognitive model and a co-evolutionary computational model they have developed. Their computational model assumes two parallel search spaces, the problem space and the solution space, both of which evolve during the design process.

To date, descriptive models have mainly focused on generating hypotheses based on observations of designers without devising experiments to test these hypotheses. These models, however, have strong advocates. For example, Gero asserts that design systems must be based on human design processes. He argues:

"Design paradigms based on mathematical models inherit the properties of the mathematical models on which they are based. Thus, it is possible to prove such characteristics as feasibility and optimality about a resulting design. However, such a design paradigm has limitations in two major areas: the processes used to achieve designs are far removed from the way humans carry out this process; and much of design can only be represented symbolically but not mathematically" [Gero 1985].

2.1.2. Synthesis and TRIZ

Synthesis is an integral part of any design activity and is recognized as one of the most important elements of the design process. In this sub-section three approaches to creative synthesis are discussed: *morphological synthesis*, *brainstorming* methods, and

TRIZ. The reason for choosing these three approaches is that each of them represents a school of thought regarding creativity. Other approaches not discussed here include literature search, emulation of nature, emulation of technical objects, consulting the experts, mind mapping, excursion, and checklists ([Pahl 1988], [Higgins 1996], [Strawbridge 2002]).

Morphological Synthesis

The morphological approach, mainly credited to Zwicky ([Zwicky 1948, 1969]), represents a philosophy that aims at changing the process of creativity from an intuitive process to a systematic one. Morphological creativity is a process for creating new ideas through analyzing the form and structure of the existing ones and changing the relationships among their components. The process of morphological creativity consists of the following steps:

- Definition: objectives and constraints of the problem are clearly identified.
- Abstraction: the related elements of the problem are clustered into groups.
- Solution: several solutions for each group are found. This may involve decomposition.
- Morphological Synthesis: the problems and their solutions are put in a *morphological matrix*. Many overall solutions are then synthesized through different combinations of sub-solutions in an unbiased operation.
- Pruning: since solutions may be too many, the ones that violate constraints or do not perform well with respect to the most important criteria are filtered out.
- Evaluation: the remaining solutions are evaluated and the best ones are selected.

Though helpful, this method has some drawbacks. The first issue is the division of the problem into its main sub-problems because usually there are no robust guidelines for performing this operation and finding the optimal division scenario. The second issue is that this method assumes that the relationships among sub-problems can be temporarily suspended, thus sub-problems can be solved in isolation. In practice, especially in mechanical engineering design problems, the compatibility of these solutions during synthesis (combination) can be problematic. The third issue is the combinatorial increase in the number of overall solutions, where most of them are meaningless. The fourth issue is that the outcome of this method is confined within the limits of ideas generated for sub-problems. That is, this method does not fully overcome the psychological inertia ([Allen 1962], [Strawbridge 2002]).

Brainstorming and Lateral Thinking

The brainstorming method was introduced by Osborn in 1957 in the United States, where the theories of Sigmund Freud were dominant [Osborn 1957]. Freud's theory differentiates between the subconscious and conscious minds. The process of ideation, or 'enlightenment', is believed to happen in the subconscious mind where thoughts are absolutely uncontrolled. The process of analysis, on the other hand, is believed to be a controlled process that happens in the conscious mind. Osborn asserted that these processes hinder each other. Therefore, he aimed to develop a mechanism to separate the above two processes in time. The only problem was to bring the original ideas, believed to be generated in the subconscious mind, to the forefront, or to the conscious mind. So he designed and conducted his famous brainstorming meetings, where everybody was encouraged to present any idea without fear of criticism. The hope was that in the storm of wild ideas the barrier would be broken and good ideas from the subconscious would come to the surface. Osborn's brainstorming method has been modified many times and several versions of it are currently available ([Hashemian 1995], [Byrne 1993]).

Edward de Bono presented a different creativity method called '*lateral thinking*' [de Bono 1992]. He drew an analogy between the process of learning and the way land is shaped by rain. In his view experiences shape our mind the same way that rainwater carves grooves in land. As more rain falls, the existing grooves become deeper and the path that the rainwater follows is no longer arbitrary; instead it traces the existing patterns of land. Similarly, as a person solves problems, his or her problem-solving experience turns into the only way to solve a problem. According to de Bono, the process of thinking (or controlled thinking) follows the routes that are already carved in the mind. Then he defines creativity as 'lateral' thinking, which is analogous to moving across grooves, or laterally, not along them. This view, however, asserts the same conclusion as Osborn's: mind practices can be improved so that a designer might access innovative ideas that are normally inhibited by the psychological momentum towards pre-existing solutions. In the basic Trial-and-Error method which will be discussed in the next sub-section, one task is to increase the number of variants considered to the greatest extent possible, and the second task is to increase the number of good variants which are based on innovative and original ideas. Brainstorming and lateral thinking help with both these tasks.

Theory of the Solution of Inventive Problems (TSIP/TRIZ)

TRIZ, a Russian acronym meaning "the theory of inventive problem solving", is a systematic approach to finding innovative solutions for technical problems. With the thawing of the Cold War climate, TRIZ entered the West over a decade ago when a few American academics began studying its principles and applying them to real design situations [Webb 2002]. The application of TRIZ enables designers to create new and improved products in a way that does not rely on the exhaustive trial of variants or accidental discoveries. Several field applications of TRIZ have been reported in the literature ([Domb 1998], [Kourmaev 2003], [Mann 2002]).

The development of TRIZ is credited to Altshuller, a Russian engineer and researcher. He observed that many processes can be controlled, at least in theory, even if they are as complicated as nuclear reactions or genetic inheritance ([Altshuller 1984], [Altshuller 1999]). He extended this observation to the process of creativity. The work of Altshuller and his colleagues started in early 1950's with the goal of developing a new technology for creativity, one that is both effective and controllable. The result was the *Theory of the Solution of Inventive Problems* (TSIP or TRIZ) and its methodical procedure, *Algorithm for the Solution of Inventive Problems* (ASIP). He presented his methods of controllable creativity in the field of technological inventions, as opposed to artistic creativity. His methods are applicable to both engineering and science.

Altshuller observed that the conventional method of human creativity is the trial-and-error method. In the absence of any knowledge, trials are performed at random. If some knowledge about the properties of variants is available, trials become more selective and the process becomes more efficient. The random trial-and-error method is not effective for complex problems of modern technology. In Altshuller's schema, the level of complexity of a problem is decided by three criteria. The first criterion is the number of trials, which is only a few for simple problems and can be hundreds of thousands for more complex problems. The second criterion is the scope of required modifications in the existing state of things; simple problems might be solved by using existing means, while complex problems might demand a change in the surrounding environment of a design object in addition to changes to the object itself. The third criterion is the scope of required expertise, which for simple problems might be confined within the boundaries of a narrow specialty, but for complex problems might extend into an entire field of technology, across other fields, or even into abstract levels of science beyond the operational technology. Noting the inadequacy of trial-and-error methods for complex problems, Altshuller stated the need for a new technology for invention that allows complex problems to be solved by fewer trials.

Altshuller and his colleagues began with the postulate that most creative solutions are generated through analogy, therefore the laws of inventive solutions can be obtained by observing previous inventions. They studied tens of thousands of patents in order to generalize conclusions about inventive solutions. They concluded that invention was to find a good solution in the solution space, and good solutions invariably solved *contradictions*. Contradictions happen when an improvement causes deterioration, when a substance needs to be liquid and solid at the same time, when we need to seal an object yet need to access it, etc. Contradictions can be eliminated by using various methods, for example by separating the conflicting elements in space and time. They noticed that despite the fact that any invention involves the removal of a contradiction, nobody systematically abstracted a problem into a contradiction in order to find a way to eliminate it. From studying many innovative designs, they also discovered that although technical inventions were many, their underlying contradictions were few. They hypothesized that objective laws must exist that can suggest efficient methods for the removal of any given contradiction. They stated that these laws could be developed through studying the methods of contradiction removal embedded in prior inventions, and that based on these laws the creative process could be systematized.

Based on the above conclusions, they developed a set of laws, rules, principles, and methods for the development of technical systems such as the ones seen in Altshuller's TRIZ Contradiction Matrix analysis and 40 principles [Altshuller 1998]. These principles have been developed further by several other researchers, improvements and applications continue to evolve, and new developments are reported in various sources such as the online TRIZ Journal (http://www.triz-journal.com). A few examples of these principles are provided below:

Law of the s-field: All solutions in the form of a technical system need to have a minimal number of elements in order to become a functional system. An object or substance (S1) requires interaction with its environment (S2) through a *field* in order to deliver its function. The field represents the energies or signals through which the technical system interacts with the outside world. This minimal system, consisting of substances (S1, S2) and a field, is called the *S-field*. The S-Field is both necessary and sufficient for the minimal description of a technical system.

Law of minimum completeness: A system must fulfill the minimum requirements that bring the system to life. If any of these requirements fails, the whole system fails.

Law of the ideal solution: The development of a system proceeds towards an ideal system that provides the function without having a system.

Corollary: A "controllable" system has at least one controllable part.

Method: Consider an ideal solution in which the underlying contradiction is eliminated without any complexity; the goal is achieved without paying any price; the function is delivered without having a machine. Then try to find a solution as close to this ideal solution as possible.

Method: use solution-neutral language to describe a problem.

Principle: Divide an object into independent parts; increase the degree of segmentation; make the object easy to disassemble.

Principle: Change an object's structure from uniform to non-uniform; make each part of an object fulfill a different and useful function.

Principle: Make an object perform multiple functions; eliminate other parts.

TRIZ proposes an algorithm for the solution of inventive problems (ASIP). In this algorithm, a problem is first abstracted into its underlying physical contradiction. Then prior solutions for this contradiction are selected from a repository of contradiction removal methods and rules. Next, a solution for removing the contradiction is chosen and is adapted to fit the characteristics of the problem at hand. This is followed by the design of the embodiment of the physical system and its evaluation and testing.

2.1.3. Axiomatic Design

Suh [Suh 1990] perceives the design process as a mapping between domains. During the design process, the problem that is being addressed can be divided into four domains: customer domain, functional domain, physical domain, and process domain. In the order listed, the elements associated with each domain are customer needs (CNs), functional requirements (FRs), design parameters (DPs), and process variables (PVs). The primary focus of the design process is the mapping from the FRs to the DPs. FRs are expressed in solution-neutral terms. By definition they are both *necessary* and *sufficient* for the description of the design goal, thus they are the minimum set of requirements which completely characterize the design objectives for a specific need. Design parameters are, in effect, solutions to FRs.

The design process progresses from a system level to more detailed levels. That is, it extends from systems to subsystems to assemblies to parts to part features. This may be represented in terms of a design hierarchy of decompositions, where each decomposition in the functional domain is performed only after a solution for a given FR is found in the physical domain. That is, the designer goes through a process of *zigzagging* between domains during the decomposition of a design problem. Although not as famous as the two axioms, Suh's statement of the nature of decomposition is in fact a very important part of the axiomatic design theory. Suh's view towards decomposition is adopted in this thesis and is utilized in the development of the design process model in Chapter 4.

After adopting the above principles for the modeling of the design process, Suh developed two axioms and several corollaries to guide the design process. His axioms are quoted below:

The Independence Axiom: Maintain the independence of functional requirements.

The Information Axiom: Minimize the information content (of the design).

As mentioned earlier, in Suh's approach a design problem is described by a set of FRs that are both necessary and sufficient for representing the goals. From the necessity of FRs, it can be logically concluded that redundancy is not allowed, therefore FRs must, by definition, be independent from each other. Then, the first axiom encourages the designer to 'maintain' this independence, and not 'couple' them through the DPs that are chosen for them. This basically means that for each FR, which by definition must be independent from other FRs, an independent DP must be found so that the desired value of the FR can be achieved through adjusting the value if its pertinent DP. The Information Axiom states that among those designs that maintain the independence of the functional requirements, the one with the minimum information content is the best solution. The design with the minimum information content is the one that has the highest probability of success, where a successful design is the one that delivers the required FRs. This design is also the one with the least complexity in terms of satisfying the functional requirements ([Albano 1993], [Suh 2001]).

Suh also introduced the design matrix, which shows the relations between the FRs and DPs at a given level of the design hierarchy. There are three possibilities for the design matrix. It can be a matrix populated both above and below the diagonal (coupled design), a triangular matrix (decoupled design), or a diagonal matrix (uncoupled design). The Independence Axiom states that the uncoupled design is ideal, and a coupled design is not acceptable and should be at least converted to an uncoupled design. Axiomatic design is not a detailed prescription on how to perform every step of the design process. Instead, it tells us how the artifact should be designed; the design should be uncoupled and with minimum information. These are general guidelines that help the designer make better decisions during the design process. Since Suh's axiomatic design is relevant to parts of this thesis, further discussions on design axioms are provided in other chapters.

2.1.4. Systematic Design

The development and advancement of the systematic design methodology is primarily attributed to the engineering design research performed in Germany ([Hubka 1980, 1988, 1992]). Some works have been translated into English, and the book "*Engineering Design: a Systematic Approach*" by Pahl and Beitz [Pahl 1988] is often considered as representative of the German approach to the systematic design methodology [Wallace 2000]. Systematic design is a truly prescriptive model of design. It prescribes a methodical procedure for design, splitting the design procedure into four main phases: preparatory, conceptual, embodiment, and detail design.

The preparatory phase involves problem definition, or the clarification of the design task. The designer must understand the customer's need and represent it as a set of goals for the design of the artifact in the form of technical requirements, which are called functions. Also during this stage the technical constraints of the design and the limitations on time, logistics, and production capabilities, as well as the evaluation criteria are identified. In the conceptual design phase, the main requirements of the problem are singled out and abstracted into a solution-neutral representation of physical functions, which reflect the exchange of material, energy, and signal among objects. Then these functions are systematically decomposed into a functional structure. The elements of the function structure are incrementally replaced by solution principles, for which conceptual physical embodiments, called *function carriers*, are developed. The decomposition and replacement of functions continue until all functions have been replaced by solutions. The systematic design approach to creativity and synthesis is mainly based on the morphological analysis explained previously. The designer is encouraged to form a matrix of all functions and their solution alternatives (the morphological matrix), generate many different combinations of these sub-solutions, evaluate these combinations, and select one (or more). The elements of a selected combination are then consolidated into the overall conceptual solution. Then the embodiment design phase begins; it is a phase that determines the shapes, arrangements (configuration), and interfaces among components, as well as the shape of the overall design. Some production and assembly aspects are also considered at this stage. At the detail design phase, the details of the product are determined and optimized. These include material types, dimensions, surface finishes and tolerances, etc. The detail design stage is completed when manufacturing drawings are produced. These design phases are iterative in nature, both within each phase and between phases.

The systematic design approach is a practice-oriented and detailed methodical procedure for performing engineering design. It is based on the postulate that benefits can be obtained if the systematic procedure is followed. Some attempts have been made to verify this postulate (e.g. [Hykin 1975] [Tebay 1984]). The actual process that designers follow has been compared with the systematic process; the results indicated that they are not the same. Also, some designers were taught the systematic process and were then asked to follow it in designing a case study object. Their performance was compared with that of a group who possessed similar skills but did not follow the systematic method; using the systematic approach did not reveal any advantage. These tests on systematic design are not conclusive, however the fact remains that researchers have not reported many success stories about the use of the systematic method [Finger 1989]. It seems that systematic design needs to be improved in order to become closer to what designers would do naturally. Also, the inconclusive experimentation with systematic design might be attributed to a lack of knowledge and experience with this process amongst designers; this is a barrier that can be overcome by rigorous training of designers in the area of the systematic design methodology.

2.1.5. Knowledge-Based Design

Knowledge-based design is a general concept and refers to design theories, methodologies, and tools that are related to the tasks of capturing, modeling, representing, comparing, and utilizing various types of design knowledge ([Coyne 1990], [Li and Zhang 1998], [Ullman 1994]). Knowledge-based design systems are typically computer software systems that operationalize the design knowledge elicited by knowledge engineers. Different types of knowledge are used during various activities of the design process, and they are modeled differently from one another. For example, during synthesis the knowledge of the prior designs (experience) is used, while during evaluation the knowledge about manufacturing processes and the life cycle characteristics of a design is utilized.

A fundamental research issue is to determine the appropriate representation frameworks for different types of design knowledge [Hashemian 1996]. Li and Zhang have classified the design knowledge into four categories: (a) artifact (product) structures, (b) artifact behaviors, (c) artifact functions and (d) causalities among structures, behaviors and functions [Li 1998]. Their classification is consistent with the function-behavior-state model developed by Tomiyama *et al* for the computer modeling of functional design [Tomiyama 1993]. Based on this classification, they developed a hybrid graph approach to represent design knowledge so that the automatic computer-based comparison of design knowledge is attained through the comparison of hybrid graphs.

Many researchers assert that 'prior experience' is the primary source of creative knowledge for designers [Kolodner 1992, 1993]. This has resulted in several *case-based design* systems for modeling the conceptual design knowledge [Hashemian 1995]. Cased-based design systems directly catalogue prior experiences and do not generalize them into rules. Several researchers (e.g. [Gero 1989-a, 1989-b]) have suggested that designers form their individual design experiences into generalized groups of concepts at many different levels of abstraction. Hashemian and Gu proposed a case-based design system in which prior design knowledge is divided into three levels of abstraction, namely solution principles, design prototypes, and parametric designs [Hashemian 1997-b]. Cases in different levels of abstraction are indexed differently and are accessed by different retrieval techniques. Their model is intended to represent the creative knowledge of designers. In some cases the expertise of the knowledgeable designers in a field can be captured and generalized into heuristic rules. This has produced numerous *rule-based design* systems for various tasks of the design process ([Rigo 2003], [Cherian 2001]). These expert systems typically replace human experts, whose participation is necessary for concurrent engineering. These systems, however, are applicable to specific problems in a given field. Li and Wu discuss the generalization of design knowledge [Li 1998]. Generalization can be described as the process of taking a large number of design examples, then extracting and retaining the salient properties in the form of conceptual descriptions (concepts) that can assist future design activities. Generalization includes three aspects: a knowledge representation for generalized concepts, a description language for design examples, and generalization operators that extract concepts from examples. Knowledge-based systems can be classified into three categories: model-based, rule-based, and case-based [Coyne 1987, 1990].

2.1.6. Decision-Based Design

Designing involves making decisions at almost every step of the process, generally in the form of choosing from among several alternatives. The process of design is viewed as decision making by many researchers (e.g. [Mistree 1990], [Hazelrigg 1998], [Olewnik 2003], [Wassenaar 2001]). Design activities consist of two processes: synthesis of solutions, and evaluation of the generated alternatives to find the best one. The focus of decision-based design (DBD) is the latter. DBD is a methodology that uses the rules of decision theory and its related sciences in design. Related headings in this area of science include Utility Theory, Multi-Attribute Decision Theory, Game Theory, Information Science, Analytical Hierarchy Process, etc. Decision sciences and their application in design, or DBD, have been extensively discussed in the literature (e.g. [Luce 1957], [Tribus 1969], [Hazelrigg 1996]).

From an abstract point of view, an important aspect of DBD is that it replaces the artifact-oriented design process with a decision-oriented process. In design problems that involve more analysis and calculation than synthesis and decision-making, an artifact-oriented approach is appropriate. However, in more decision-intensive situations the artifact can not represent the history of risk and uncertainty associated with the decisions. Paying direct attention to decisions (instead of the artifact) reduces the chance of making a wrong decision. Thus, DBD reduces the number of design iterations, which are invariably unwanted, by paying greater attention to decisions throughout the entire design process, and by considering all the factors related to decision-making including life cycle issues. Also, DBD helps designers retain the design rationale which is captured in the form of decisions and their reasoning. This is especially useful in redesign tasks, where prior experience is often utilized.

Design decisions are usually made based on multiple criteria. These criteria are often in conflict with one another (personal preferences, resource requirements, etc.). DBD offers tools and techniques to deal with conflict resolution and multi-objective optimization [Chen 2002]. It also offers methods for the inclusion and organization of a large number of life cycle criteria in the evaluation and optimization of design alternatives [Yoshimura 2003]. Usually some of the attributes of a solution are qualitative. DBD suggests methods for the quantitative measurement of qualitative parameters; an example is the systematic function-based evaluation of design alternatives [Iyengar 1994]. It also offers guidelines to deal with ambiguity, risk, and uncertainty that are a part of human decision making and human subjective evaluations. The decisions may be made individually or collaboratively; DBD also proposes approaches to distributed decision making [Jeong 2002]. Further, designers rarely know the entire set of potential alternatives, nor do they know all the evaluation criteria and their relative importance; in most design problems both the solutions and the evaluation criteria evolve as work progresses [Maher 2003]. In design problems, often there is not enough time to obtain high levels of knowledge, thus a final decision has to be made with incomplete knowledge. Making decisions with incomplete knowledge is another area of research in DBD [Pender 2001].

2.1.7. The General Design Theory

The General Design Theory (GDT) was advanced by Yoshikawa (the English presentation [Yoshikawa 1981]). GDT is a formal mathematical theory of design. It models the design process in the framework of set theory. GDT also provides a prescription for the development of CAD systems. This sub-section will not discuss the details of the definitions, axioms, and theorems of GDT. More information about the theory can be found in [Yoshikawa 1981] and other related papers (e.g. [Tomiyama 1994], [Tomiyama 1996]).

GDT starts with assumptions about objects in the world and uses these assumptions to prove theorems about the nature of design. Yoshikawa considers design as a “typical” intellectual activity that humans perform. He considers designing ability intrinsic for human beings and states that design knowledge and skills are developed unconsciously by repetitive experiences. He then sets the ultimate aim of his general design theory as the clarification of the human design ability in a scientific way. This view is essential to his theory and is briefly explained below.

He models the world as a set of entities (e.g. an apple) and their attributes (e.g. color). He asserts that humans construct abstract concepts about the classification of objects of the real world. People have the ability to construct concepts about various characteristics of entities and classify the natural entities according to these conceptual characteristics. Each classification is a sub-set of the entities of the world (e.g. all sweet objects) and includes the elements that share the same characteristics. A characteristic represents a value, function, etc. (e.g. taste).

Once classification is done, people can identify entities not only by their direct image (attributes) but also by using their characteristics, that is, by using the classification (sub-set) they belong to. Many entities belong to several classifications at the same time; they can be identified through Boolean operations on these classifications. Combining these classifications can result in concepts that do not have any correspondence to a real world, but may have a higher value than any existing entity. Yoshikawa asserts that such "*conceptual combination of abstract characteristics is the necessary condition for invention*".

Yoshikawa then states that the necessary condition for designing, that is, the act of creating artificial things which do not exist in the real world, is the formation of concepts of non-existing entities as the result of performing logical operations on knowledge about existing things. His view of this logical operation is explained in the previous paragraph. Designing through the logical operations on the conceptual sets is the cognitive foundation of his theory. He then presents several definitions and axioms and proves several theorems based on his assumptions.

Yoshikawa's approach is mathematical and involves assumptions of an ideal situation that may differ from the actual design practice. For example it utilizes the concept of continuity which guarantees that a small change in the design description will result in a small change in the artifact functionality and vice versa. GDT is a descriptive model, but it also provides some guidelines (e.g. appropriate data structures) for the development of CAD systems. Reich states that one concrete conclusion of GDT is about the way attributes must be presented in CAD systems. According to GDT, attributes are defined by the entities that have them. This is not as intuitive as the more common method of defining an entity by the list of its attributes [Reich 1995].

2.2. Review of Product Configuration Design Research

Mechanical design methods which are related to AD can be sought in product *configuration design* research, which encompasses such topics as modular design, product family and platform design, design for mass customization, interface design, etc. The configuration design stage refers to the determination of a product's *architecture*. Architecture includes the layout of physical subsystems, their functions and embodiments, and the overall shape of the product [Pahl 1988]. The configuration design is not treated as a separate stage in the design process by some researchers (e.g. [Sydenham 2004]). In this case the activities that comprise the configuration design stage are embedded or implied in other stages of the design process. The reason is that the embodiments of assemblies and components and their layout in a product are often a by-product of other design activities, and the product's configuration is not explicitly designed. This is despite the fact that many life cycle performances of a product are influenced by the way its assemblies are organized. In order to materialize the potential benefits of product configuration design, two elements of a product's configuration must be considered: the *physical structure* and the *functional structure*.

2.2.1. Functional and Physical Structures

The original functional requirements of a design problem are invariably decomposed during the design process, resulting in a hierarchy. This hierarchy represents the design rationale. It shows how functions are decomposed, what decomposition scenario is selected, what auxiliary or technical functions are introduced at each level, and at what stage the decomposition of a function ends (Figure 2.1-a).

The *end-node* functions are directly realized by independent physical parameters without further decomposition. Functions within the hierarchy are also related to each other through functional interactions, which are various logical and physical relationships in the exchange of material, energy, and signal [Pahl 1988]. These relations can be represented in a graph (Figure 2.1-b). The *functional structure* is a representative of both the design rationale (decomposition hierarchy) and functional interactions (graph) (Figure 2.1-c).

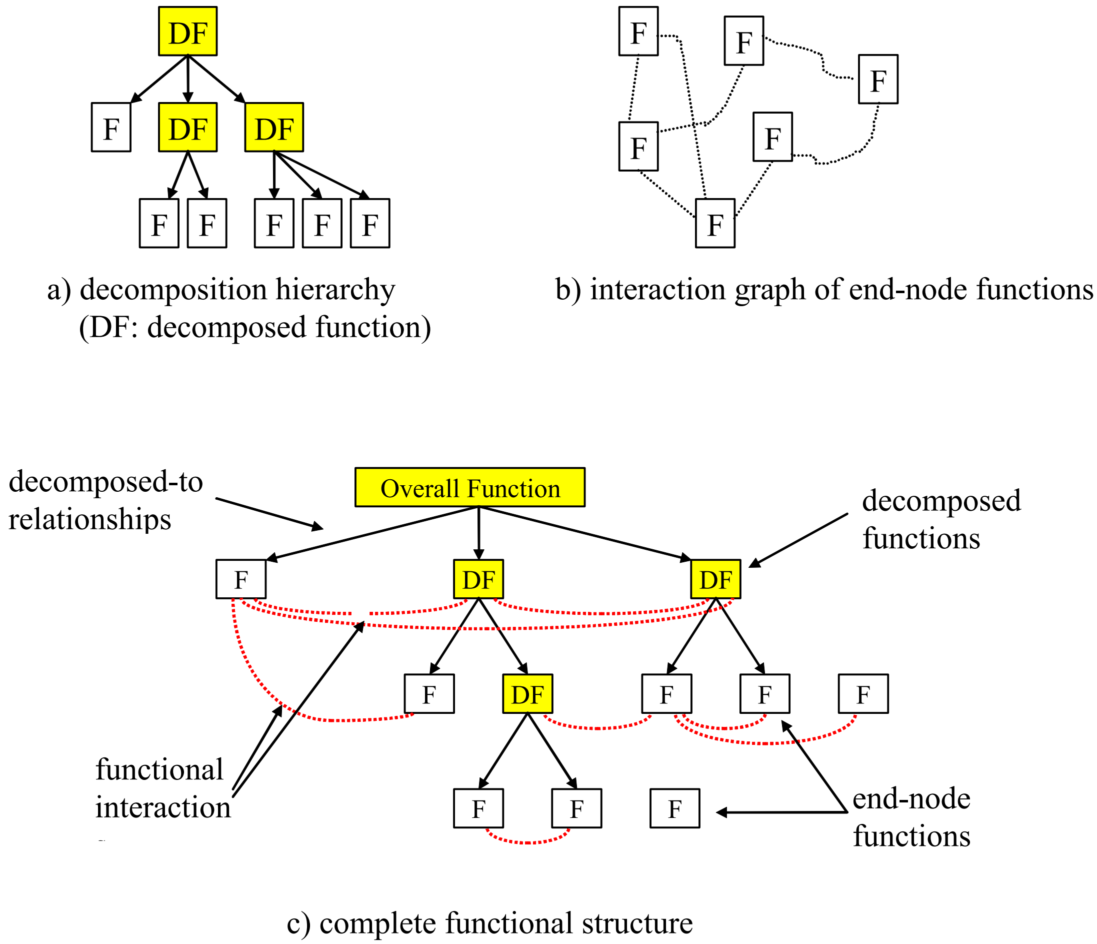


Figure 2. 1: The functional structure.

The *physical structure* refers to the hierarchy of physical assemblies, subassemblies, and components. The physical structure includes the overall layout of assemblies and components, as well as various relations that exist among them. Similar to the functional structure, these relationships can be represented by a graph.

Therefore, in conventional mechanical design the configuration design of a product is often subservient to decisions made during conceptual and detail design stages. That is, the hierarchy of assemblies and their layout are typically designed in such a way as to assure the proper functioning of the product. The departure from this practice can be seen in design methodologies that treat product configuration design as an explicit design activity. For example modular design develops assemblies and their layout with the objective of achieving further benefits with respect to the product's life cycle [Gu 1997].

The adaptable design presented in this thesis also extends from conceptual design to product configuration design; it involves the determination of the overall architecture of products. The objectives of product configuration design can be categorized as relating to the physical functions of the product, to aesthetic or ergonomic factors, or to life cycle factors such as production, serviceability, and recycling [Gu 2004]. The following sections review some research approaches that aim at the explicit design of the configuration of mechanical products.

2.2.2. Modular Design

Modular design is a design methodology that aims at developing a product architecture consisting of distinct sub-systems in order to achieve a set of perceived benefits ([Gu 1999], [Marshall 1999]). Modular products fulfill various overall functions through the combination of distinct building blocks or modules [Hillstrom 1994]. If the common practice in a domain is to design integrated products, the application of modular design requires substantial changes to the existing design procedure. For example, modularizing a car into separable modules or an engine block into separable cylinders requires a departure from the current design practice. If, however, products in a domain naturally consist of distinct sub-systems, modular design is simply considered to be a good design practice. Examples include the design of detachable speakers for stereo systems or detachable gearboxes for electro-motors.

A modular product, unlike an integrated product, has its components clustered into distinct sub-systems (modules) so that these modules can be designed, manufactured and assembled separately. Modules can be physically detached from the overall product to be repaired, recycled or upgraded. They may be used in other products that utilize the same or similar modules, or may be arranged in different configurations to obtain several functions from the same overall product. The modules of a product, depending on its size and complexity, may in turn have their components clustered into smaller modules to form a hierarchical modular structure.

It is difficult to precisely classify products into modular and non-modular because various levels of modularity may be observed in the construction of products. Ulrich *et al* observed that good modular designs usually exhibit some common characteristics [Ulrich 1991]. First, a modular product is constructed by a set of compatible basic units, or modules, which can be used to construct a variety of other products. These units can be general units that are used in various products, or specific units for particular models. They can be introduced during the initial design, or they can be introduced as new technologies become available or new demands arise. Second, the interface among units must allow for simple assembly and disassembly, and must be compatible and consistent across various relevant units. Third, a modular product is constructed in two phases. The first phase is the design and production of basic modules. These modules are general and are not specific to a single product. The second phase is the design and construction of completed modular products using the basic modules developed in the first phase, and if required, other parts or procedures specific to the product. The second phase may be undertaken by the producer of the basic units, or by other parties. This discussion reveals that the two fundamental issues of modular design are the development of modules (*clustering* of components), and the development of *interfaces* among them.

Clustering (Segmentation)

Most modular design approaches provide methods for finding the optimal scenario for the clustering of components of a design. Gu *et al* have presented a multi-objective optimization method for the clustering of components for diverse life cycle objectives ([Sosale 1997], [Gu 1997]. In their approach, the relationships among components are mathematically represented by an *interaction matrix*. The quantitative values of cells of this matrix indicate how strongly two components (rows and columns) must be in the same module based on various life cycle criteria called perspectives. For example: for the objective of recycling plastic components should be placed in the same module, while for the objective of repair frequently failing parts should be clustered in the same module. They use a genetic algorithm method to find the optimal scenario with respect to several objectives that may be in conflict with each other. The use of genetic algorithm simplifies the combinatorial optimization problem, as discussed in [Kamrani 2003]. Yu *et al* have developed a similar approach using genetic algorithms for the partitioning of a product into an optimal set of modules, and have applied their clustering method to the modular design of an industrial gas turbine [Yu 2003]. Various approaches to the clustering of components usually include some or all of the following general steps of a basic modular design methodology:

- Establish the functional structure of the product.
- Determine the solution alternatives adopted for each function.

- Establish the life cycle objectives and determine their relative importance.
- Decide on the level of abstraction in referring to design components.
- Build the relationship matrices and quantify their elements (interaction analysis)
- Execute algorithms to find the optimal clustering scenario.
- Modify shapes and interfaces to comply with the proposed product architecture.
- Modify the design data (interaction matrix) based on new relations between modules and iterate the execution of algorithms.
- Evaluate the improvement of the final design with respect to the objectives of modularity.

Interfaces

Gu and Slevinsky have discussed the design of interfaces for the assembly of modules in the overall structure of modular products [Gu 2003]. They emphasize that the connections between modules must be designed to facilitate the separation and attachment of modules for parallel manufacturing and assembly, as well as for performing post product life activities. They present the concept of *Mechanical Bus* for facilitating modular and platform product design, and identify the characteristics and features of mechanical bus interfaces. These include bus functions, locking and release mechanisms, positioning and locating features, and so on. Sanchez emphasizes that the standardization of interfaces enables developers to produce the latest components or devise processes compatible with a company's existing products and processes [Sanchez 2002]. Such standardization of interfaces helps in using the same modules across a portfolio of products. This results in savings for the producer because members of a product family not only share design similarities, but also share process plans, parts, assembly lines and other manufacturing infrastructures [DeLit 2003]. Hillstrom presented a method that helps the designer clarify how interfaces between modules influence module functions, and select the best interface location. His method utilizes Suh's axiomatic design theory together with conventional DFMA tools (Design For Manufacture and Assembly) [Hillstrom 1994].

Discussion

Most current modular design approaches aim to develop modules using *component-based clustering*. It is assumed that the interactions among various components can be quantified, and thus can be represented by graphs or matrices. Despite the great benefits of these techniques, there are also some drawbacks:

First, there is uncertainty about the values or numbers provided in interaction matrices and they are typically determined by estimation. If several matrices are superimposed via weighting factors, the numbers in the final matrix can significantly change when different designers prepare the matrix. The data inside the matrix is fed into a clustering algorithm. Depending on the sensitivity of the algorithm to the initial data, the final results may vary substantially, resulting in low repeatability and hence low reliability of the method. Reliability of the clustering method can be improved by applying stricter discipline in data collection and refinement, and by the ‘preparation’ of the initial matrix using data analysis methods such as the Analytical Hierarchy Process, and using algorithms that are less sensitive to variations and errors in the initial data [Sand 2002].

Second, the clustering problem has combinatorial complexity. The problem of clustering *n* components is equivalent to the problem of partitioning an n-element vertex set. The number of ways the set can be partitioned cannot be calculated by a simple formula and is a non-polynomial function of n [Stanley 2001]. For every candidate scenario, a calculation of the objective function based on the interaction matrix is required. Therefore, an exhaustive method to find the best scenario is computationally impractical as n increases. Heuristics rules and non-deterministic optimization methods have been utilized for the clustering problem; examples include genetic algorithms discussed earlier, and fuzzy cluster identification [Tsai 1999].

Third, the optimal clustering of mechanical components for several life cycle objectives may result in the development of product-specific modules that do not perform obvious functions. These modules are very context-sensitive and cannot be used in a variety of circumstances. This characteristic is particularly important for adaptable design, where various applications usually require functional reconfiguration more than structural reconfiguration. Hillstrom addressed this issue by including the functional structure data to clarify the interface/function interactions, then using these interactions to determine clustering of components and optimal locations for interfaces [Hillstrom 1994].

Fourth, these approaches often ignore the designer's freedom to change the values inside the interaction matrices. These values are based on the specifications and properties of physical components; for example material type is a property that determines the interaction value with respect to the recycling objective. The designer often has freedom to change these properties, or even to change the solution principle for a particular function at the conceptual level. One approach to addressing the design freedom is to first find the interaction data within the matrix that are critical to the clustering algorithm, and then replace the components that affect the data with alternative solutions. The computer implementation of such a method is possible if the functional and physical structures can be represented simultaneously (Please see Appendix 1 and the pilot software TARRAUH). Then, the physical structure determines the values inside the interaction matrices and the functional structure allows the designer to choose alternative solutions from a library of design cases, where cases are indexed according to their functions [Hashemian and Gu 1997].

Despite these shortcomings, modular design has been the most successful approach to product configuration design. Various field applications of modular design methodologies have been reported in the literature. For example, Cohen *et al* applied a modular design method to the conceptual design of a modular robot in order to achieve hardware flexibility, as opposed to the conventional methods that rely on software flexibility [Cohen 1992]. Their strategy is to develop an inventory of basic modular units, standardized and minimized in size, which will allow the user to configure the most suitable robot geometry for a given task. Thus, modular robots with hardware flexibility yield optimal arm geometries. This gives them an advantage over conventional industrial robots that are used in flexible automation.

2.2.3. Product Family and Platform Design

Industry's response to rising consumer expectations has been an emerging production paradigm called *mass customization* ([Pine 1993], [Tseng 1996], [Rai 2003]). This paradigm differs from the conventional mass production, which involves the manufacturing of many identical products using large assembly lines. The approach towards mass customization adopted most by companies is the development of platform-oriented multiple product variants. Various design strategies such as *design for* *variety* and *product family design* have become critical in this respect ([Rai 2003], [Martin 2002]). A platform-based design strategy is commonly used to create a product family for variety and mass customization.

Platform design is a specific type of modular design in which the objectives of modularization relate to multiple products that are considered members of the same family or portfolio. Platform design is based on the identification of common attributes within the members of a product family. Many companies are utilizing product families to increase variety, shorten lead-times, and reduce costs. Simpson states that the key to a successful product family is the product platform from which it is derived, either by adding, removing, or substituting one or more modules to the platform or by scaling the platform in one or more dimensions to target specific market niches [Simpson 2003].

Du *et al* investigate the fundamental issues underlying product family development. They introduce the concept of Architecture of Product Family (APF) as a conceptual structure for the overall logical organization of a family of products. In their work, APF constructs, which include common bases, differentiation enablers, and configuration mechanisms, are discussed from both sales and engineering perspectives. Also discussed are variety generation methods with regard to producing custom products based on modular product architecture and configure-to-order product development. To support APF-based product family design, a Generic Product Structure (GPS) is proposed as the platform for tailoring products to individual customer needs and generating product variants [Du 2001].

Members of a product family may have similar overall functionality, or they may share only some of the same functionality. However, to some degree even seemingly unrelated products can be built on a common platform [Gu 2004]. In the general sense, a platform is any set of standardized parameters, which are maintained within a group of products for compatibility ([Farrell 2001], [Simpson 2001]):

- Process and Manufacturing Platforms – In this case the platform is not so much integrated into the design of a product family; it is integrated into the process of how the products are manufactured.
- Component Standard Platforms – Component standard platforms unify manufacturing issues in multiple products by using common components whenever possible.
- Modular Platforms – This type of platform makes use of modules between several products so that common parts are used whenever possible [Gonzalez-Zugasti 2000].

The methods and guidelines for the development of platforms and modules involve the study of similarities among various products within a portfolio ([Gu 2003], [Zha 2002], [Du 2001], [Tsai 2004]). The optimization of the modularity scenario can also be performed with respect to other life cycle objectives ([Ishii 1994, 1995, 1998], [Gu 1999], [Yu 2003]). In mass customization, development of customizable features is guided by the study of customers' expectations. Tseng *et al* have suggested *design by customers* for mass customization [Tseng 1998]. In their work, the design and manufacturing capabilities of a company are represented in a product family architecture, using which customers assert their needs and define variations from base products. Issues related to product family development are also discussed in [Du 2001], [Yu 1999]. They present the architecture of the product family as a conceptual structure and the overall logical organization of constructs: base, differentiation enabler (module), and configuration mechanisms. Such a modular architecture provides for configure-to-order product development, tailoring products to customer needs and generating product variants.

Simpson *et al* proposed a model to design a product family based on the concept of a scalable platform, which can be sized to provide necessary variants [Simpson 2001]. Otto *et al* presented an optimization-based method for designing product platforms that takes into consideration both the technical performance requirements and the cost of the product family. They define a product platform as the set of subsystems shared across the products that are offered by a firm. They point out that there are two possible ways to create a product family using a platform-based strategy: integral platform, in which individually designed assemblies are attached to an integral platform shared among all products to create different variants of the product family; and modular platform, a strategy that uses a set of modules as a product platform that is used across the product family. Then, the common modular platform along with other variant-specific modules is used to generate the product family variants. They mention the advantages of the modular platform strategy: reducing the number of module types and instances of each type lowers the efforts required to design, produce, distribute, operate and maintain the module instances throughout the life cycle of the product family ([Gonzalez-Zugasti 2000], [Dahmus 2001]).

2.2.4. Life Cycle Objectives of Modularity

Several life cycle objectives or benefits can be sought in the modularization of a design. The following is a list of most important objectives of modularity, developed by Gu *et* *al* [Gu 1997].

1. Splitting the Task for Parallel Product Development

Sequential development of a complex product may take an unreasonably long time; therefore the overall task is decomposed into sub-tasks. This allows each sub-task to be handled independently and performed in parallel with other tasks. Design of complex products may involve several design teams consisting of several experts across different fields. The design of a product can be modularized based on such splitting of the task. As a result, a complex and lengthy project can be reduced to several smaller, focused and specialized projects that can be performed simultaneously. Eppinger discusses the use of matrix manipulation techniques for the partitioning of design tasks [Eppinger 1990]. In his approach the rows and columns of the data structure represent design tasks. The data structure is a binary matrix, where the unity indicates that the task in the row requires the information from the tasks in the column. The matrix partitioning is done by rows and column swapping techniques to create a diagonal or a triangular matrix. Issues related to this objective of modularization include:

- Identifying key factors which determine the fate of the design process.
- Reducing both perceived and actual complexity of design tasks.
- Sharing engineering data at an earlier stage of product development.
- Redefining the critical components.
- Forcing the designers to organize their decision process.
- Coordinating the different tasks.

2. Manufacturing

Higher production volume reduces the per-unit production cost. Through the development of standard modules that can be shared among various models or among products within a portfolio or product family, the manufacturer can increase the production volume for identical units. This results in reduced design and production effort, reduced inventory, and using the same production line set up and manufactured component for different products. Further manufacturing benefits of modularity include:

- Dividing a product into independent components allows design and production activities to be specialized and focused.
- Component sharing allows higher initial investment in production resources because these higher costs are distributed across a large number of units.
- Acquisition of improved production technology with low unit manufacturing cost is possible.
- The quality and reliability of products can be improved, while the production lead time is reduced.

3. Assembly

The purpose of modular design for assembly is to modularize a product (only one, not a variety of products) into sub-assemblies that can be produced and assembled in isolation. These sub-assemblies are then assembled together to build the overall product. This type of modularity can be of paramount importance for large and complex designs that may involve considerable assembly time.

4. Repair and Maintenance

Modularization creates segmentation in the structure of a product. Therefore, when the product fails, diagnosis can be performed methodically as modules can be examined individually. Once the faulty module is identified, then its components can be examined and individual components can be repaired or replaced as necessary. The replacement of modules and components is simplified as a modular structure generally facilitates disassembly. Therefore, modularization generally helps with fault detection in real time and reduces down time. Similarly, modularization can help with the maintenance of a product through systematic maintenance procedures and easy access to various modules and components. Modularization may be performed explicitly for the benefit of repair and maintenance. For repair, frequently-failing components can be grouped in a same module; in the event of failure this module is replaced with a spare one and the repair can be performed while the product stays in operation. For maintenance, the components that require the same frequency of service may be grouped together, allowing for easy organization of maintenance schedules.

5. Upgrading and Renovation

Today's highly competitive market and high consumer expectations demand manufacturers to introduce a variety of new models in short periods of time. Often a new model has to reach the market quickly, and may not permit sufficient time for the design of the new model from the beginning. In such circumstances only a redesign is possible. Also, if there is not adequate time for a new manufacturing set-up, the existing production layouts and even component inventory have to be utilized. Modularization allows companies to introduce new models to the market quickly and inexpensively as only a few modules have to be redesigned and reproduced, while many modules, platforms, and process plans can be reused from an earlier design.

6. Product Variety

Various products frequently share similar functions among their constituent components. Therefore, it may be possible to develop standard modules for these common functions. Consequently, a variety of models may be easily developed through the combination and configuration of these standard modules, and if necessary, any model-specific modules. This benefit is the most common motivation for modularization and is often implemented in the form of platform design and add-on modules.

7. Recycling and Reuse

Different components of a product are made from different materials and require different processes for their recycling. The clustering of components with similar recycling properties into a single module greatly facilitates recycling as it makes it possible for a large module to be recycled without any disassembly of its components. Also, the service life of durable components within a product or the components that do not undergo much wear and tear can be substantially longer than the service life of the overall product. Such components can be salvaged and reused when the product is recycled or disposed of. The clustering of long lasting components in the same module increases the possibility of reuse and requires less disassembly for part recovery.

8. Customization

Consumer tastes are often highly diverse. This complicates the design and manufacturing of a product that is satisfactory to all customers. Modularization is a very efficient technique for product customization. Via the rearrangement, swapping, replacement, and addition of common modules, as well as using specific modules developed for specific product demands, the producer is able to develop various product features and functions for mass customization or make-to-order customization.

2.3. Discussion

Configuration design approaches discussed in this chapter do not make a reference to design for *adaptability*. Instead, they are concerned with particular benefits of product configuration design such as the development of product families for mass customization or the development of modular products for life cycle objectives. The idea of *adaptable design* as a new design paradigm for the development of engineering products was proposed by Gu [Gu 2002]. The important difference between adaptable design and the existing product configuration design approaches is that in the AD paradigm *adaptability* of a design is treated as a distinct design characteristic.

The next chapter discusses various types of adaptability in detail. These types of adaptability are applicable to a specific physical product, to a design from which various products can be developed, or to both. They also include both the adaptations that happen over the course of time and the adaptations that extend the scope of application for a product or for its design. Such an extended concept of adaptability includes the existing approaches towards increasing *variety*, *customization*, *upgrading*, and even *versatility* in engineering products. Several of these approaches were reviewed in this chapter. They all aim at increasing the adaptability of a design for specific objectives known at the beginning of the design process. This thesis extends adaptable design to include *general adaptability*, in which specific objectives are not targeted *a priori*. Further, all these approaches are based on the modularization of designs, whether it is for life cycle performance of a single product or for the reduction of production cost via sharing product platforms and modules. The general adaptability proposed in this thesis uses *functions* as the main criterion in the modularization process, resulting in a new product architecture in which every product is a temporary assembly of functional modules.

Chapter 3: Adaptability in Designs and Products

The ultimate goal of AD is to extend the utility of products and their designs. Utility of a product can be extended over the course of time, or it can be extended in the scope of applications; adaptable design can facilitate adaptations to foreseen changes or to unforeseen changes; adaptations may occur for a variety of reasons such as the upgrading of obsolete components or the customization of functions; and adaptations can be applied to a physical object or to a design (blueprint) from which a variety of products can be developed. This section discusses these concepts.

Section 3.1 discusses the environmental aspects of this research and presents a view in which adaptation is regarded as the extension of the utility of artifacts. Section 3.2 utilizes this view to categorize various types of adaptabilities related to engineering design. Section 3.3 describes the characteristics which make a design suitable for adaptability. Section 3.4 discusses the benefits of adaptable design for the user, the producer, and the environment. Section 3.5 summarizes the chapter.

3.1. Extension of Utility

This section describes the environmental implications of adaptable design. Section 3.1.1 discusses some existing assertions that the environmental problems are critical and therefore the inevitable trend of market is towards a *service-based economy*. This term refers to a marketplace in which the scarcity of resources limits the volume of objects we can produce, thus the purpose of engineering becomes the utility of objects not the objects themselves. Section 3.1.2 discusses the *extension of utility* as a viable response to the scarcity of resources, and views the adaptation of a product as one way of extending its utility. Section 3.1.3 summarizes these discussions and concludes a logical relationship between scarcity of resources and the need for AD.

3.1.1. Service-Based Economy

Recent decades have increasingly witnessed the depletion of natural resources and the pollution of environment. The seriousness of environmental problems has become more evident by such phenomena as global warming, which have raised questions about the sustainability of the current global development ([Meadows 1992], [Mebratu 1998], [Bhaskar 1995]). Various social, political, economical, agricultural, and industrial practices contribute to environmental problems. These parameters are often intertwined and a valid solution for environmental problems inevitably requires changes in almost all of these practices. The scope of this discussion, however, is limited to the environmental aspects of *engineering production*.

Increasing population and the daily dependency of modern societies on various products have resulted in unprecedented engineering production volumes. Every year billions of tons of virgin materials are transformed into various products, consuming energy and producing waste and pollution in the process. These products eventually retire and for the most part are discarded at landfills. Thus from the environmental point of view ‘engineering production’ is the transformation of natural resources into waste and pollution. There are concerns that the current transformation rate is beyond Nature’s rejuvenation capacity ([Meadows 1992], [Bhaskar 1995]).

Concerns about *sustainable development* have motivated various changes in practices and processes related to the life cycle of products ([Housechild 1998], [WBCSD 1998], [Lu 2003], [Tipnis 1998], [Rifera 1999]). The life cycle of a product can be divided into three phases: production (creation), service (operation), and retirement (disposal/recycling). Reducing the environmental impact of engineering production has been sought in all three phases of a product's life cycle. For instance, production processes have improved in terms of gas emission, energy consumption, and material waste; products are designed to be more energy efficient and to cause less pollution; toxic or non-recyclable materials are replaced with more environmentally-friendly materials; and products are designed for recycling and reuse.

In addition to these life cycle considerations, there is another approach which aims to "reduce the total production volume" [Hashemian 2004]. Since maintaining the convenience and quality of life in developed societies requires a certain number of products to be in service at any given time, lower production volume means longer service life for existing products. The need for *more service with fewer products*, primarily caused by the scarcity of resources, increases the tendency towards a *service-based economy*, which is believed to be the market trend of post-industrial societies ([Tomiyama 1995], [Seliger 1997], [Tomiyama 1997], [Seliger 1998], [Tomiyama 1999], [Tomiyama 2000], [Fujimoto 2003]).

The premise of a service-based market is the dematerialization of services, assuming that services cause less environmental harm than physical products do ([Oksana 2000], [Brezet 2001]). From a practical standpoint, the dematerialization of services means obtaining more usage from fewer manufactured products, that is, using products to their fullest potentials. The increasing of the usage of products is called the *'extension of utility'* in this thesis. Also, service-based market represents a utilitarian view towards engineering. In this view an artifact is created for the purpose of performing a *function* and not so much for aesthetics or other properties related to the physical object. This argument will be utilized in Chapter 4 to propose the 'frame-and-function' architecture, in which a product is constructed as a 'temporary assembly' for performing a function. In this architecture, only the function is important and the product itself will be dismantled or reconfigured for new usage after its current function is no longer needed.

The next section discusses the extension of utility and its relation to designing products for adaptability.

3.1.2. Extending Utility through Adaptation

The utility of a mechanical product can be extended in two ways: by prolonging its service life in its normal operational mode, and by adapting it to new operational modes. The first method is applicable when the need for the current function remains longer than a product's life; the second method is applicable when the product's current service becomes unwanted while the product is in good working condition.

Extension of Normal Service: *Durability*

Products retire when they can no longer satisfactorily deliver their expected functions due to deterioration, damage, wear, aging, etc. The service life of these products could be prolonged either by "designing out maintenance" or by "designing for maintenance" [Markeset 2001]. Design of maintenance-free products is influenced by quality, reliability, and durability characteristics on one hand and by cost and technological limitations on the other hand. Design for maintenance requires risk analysis, study of possible failure scenarios, and making provisions for easy disassembly and repair of frequently-failing parts [Gu 1999]. Regardless of maintenance issues, in this thesis the prolonging of normal service is called 'durability'. Durability extends the utility of products only in their current operational mode. It becomes irrelevant if the termination of service is not due to deterioration but due to changes in service requirements.

Extension of Utility to New Services: *Adaptability*

Many products are discarded while in good working condition because their normal operation is no longer desired. The most common reason for such premature retirement is technological obsolescence. Millions of products, especially those containing electronic parts, are retired annually due to their rapid obsolescence and the proliferation of new models. Other reasons include changes in the needs or expectations of the user, changes in operation regulations, and changes in the working environment. In these cases, the utility of a product can be extended only if it can be adapted to the new service requirements. Therefore, the process of adaptation can be viewed as the extension of the utility of an artefact to new services.

Extension of Utility within Environmental Approaches

Figure 3.1 classifies various approaches towards reducing the environmental impact of engineering production. These approaches are divided into two categories: improving product life cycles, and reducing production volume. The first category includes the conventional methods applicable to the three phases of a product's life cycle. The second category leads to the extension of utility via durability and adaptability as discussed earlier in this section.

In the categorization depicted in Figure 3.1, adaptability belongs to the methods of *'reducing production volume'*. However, it is also logically close to the methods of '*recovery'* among conventional life cycle approaches because both recovery and adaptability create 'new usage' for products. Therefore, recovery methods (*recycling* and *part reuse*) are sometimes called adaptability in the literature [Willems 2003]. In this thesis, however, adaptation does not refer to material recycling and part reuse; which do not reuse several components of a product in a new operational mode. This is based on the three limits which this thesis has adopted for the definition of adaptability (Chapter 1). The differences between recovery and adaptability will be further clarified later in this chapter when the environmental benefits of AD are discussed.

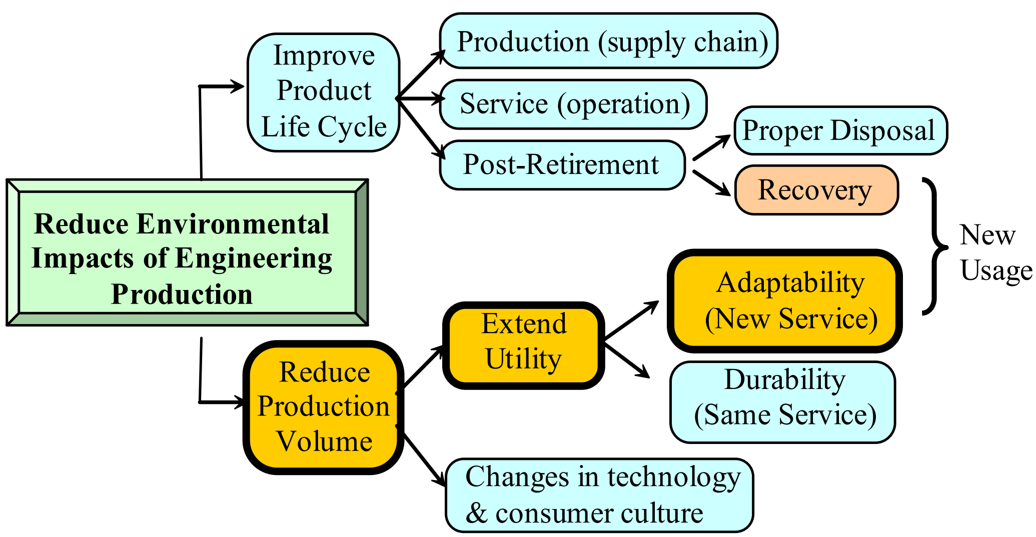


Figure 3. 1: AD within the spectrum of environmental approaches.

3.1.3. Discussion

There is no general consensus on the seriousness of environmental problems. Therefore, there is no general agreement that the service-based economy will prevail. This is evident by numerous products which are designed to be company specific, and by intense competition for higher productivity and market share instead of increasing quality and adaptability. These issues have been extensively debated in the literature ([Mebratu 1998], [Hashemian 2004], [Meadows 1992], [Bhaskar 1995]). Regardless of these debates, it is logically evident that the scarcity of resources makes it unaffordable to manufacture a large number of products. If such conditions occur, the emphasis of engineering will inevitably shift from having many objects to obtaining maximum usage from fewer objects. The extension of usage of products requires them to be durable for their normal service, or to be adaptable to varying circumstances. Thus this section established a logical link between the scarcity of resources and the need for adaptable design. This discussion is summarized in Figure 3.2.

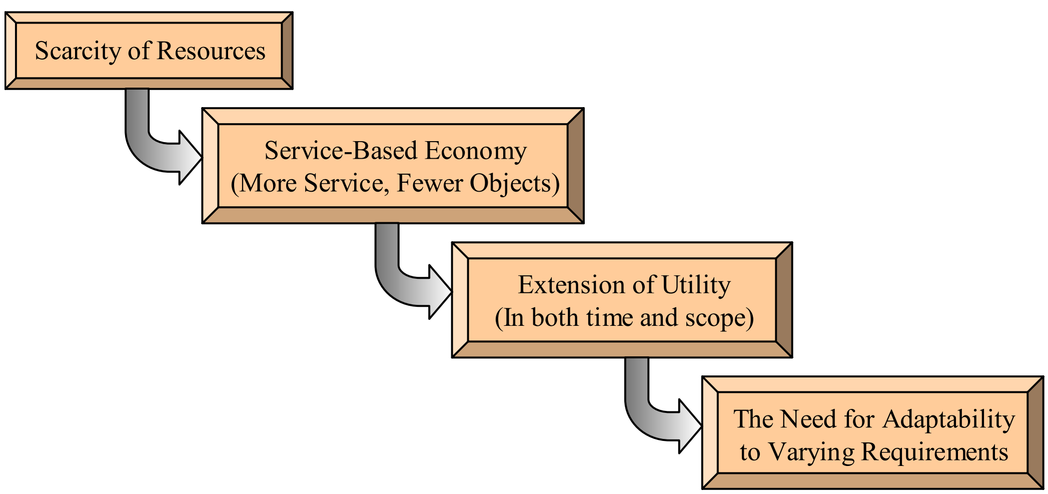


Figure 3. 2: The relation between scarcity of resources and the need for adaptability.

3.2. Categories of Adaptabilities

The adaptation of a product to new services can be viewed as the extension of utility, as discussed in the previous section. This section utilizes this view to categorize various types of adaptations. First, adaptations are divided into *design adaptability* and *product adaptability* depending on which item is the subject of adaptation. Then adaptations are divided into *sequential* and *parallel* adaptations depending on whether the extension of utility occurs over the course of time or it is unrelated to time. Then this section discusses *specific adaptability* and *general adaptability* which were mentioned in Chapter 1, and explains the four categories of specific adaptability. The section provides a summary of these categorizations at the end.

3.2.1. Design Adaptability and Product Adaptability

The design process results in the 'description' of an entity (part or product) that, when materialized according to this description, can fulfill a required set of functions. This description can be in the forms of blueprints, instructions, CAD models, prototypes, etc. Therefore, the design process and the subsequent production result in the creation of two entities: a *design* and a *product* (Figure 3.3). In 'design for adaptability', both of these two entities can be made adaptable.

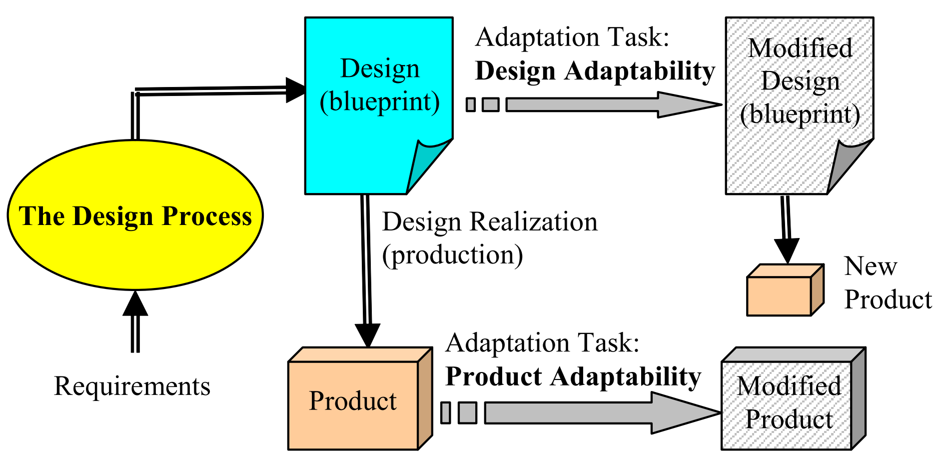


Figure 3. 3: Design Adaptability and Product Adaptability.

Design (Producer) Adaptability

A design is a set of instructions for the creation of a product or many identical products. The same design, with minor changes, can be used to create different products, usually in the form of variations of the original product within a product portfolio. Such design reuse is called *design adaptability* in this thesis. Design adaptability results in the creation of a *variety* of designs based on a common adaptable blueprint, and in the upgrading of new models through the modification of old designs.

The realization of a design into a physical product is performed by the producer; therefore design adaptability mainly concerns the producers and can be alternatively called *producer adaptability*. Design adaptability is in fact the reuse of design knowledge including proven concepts, components, methods, and processes. Thus it yields significant benefits in terms of design cost and time. Further, the adaptation and reuse of an existing design enables the producer to utilize the existing production processes and even the existing inventories of manufactured components, hence yielding further savings in production costs. Although design adaptability may not be of importance to the user who owns a single product, the design similarity between the user's product and other popular models reduces training time for operating different models, and enables users to swap components for repair and upgrading purposes.

Figure 3.4 shows two models of SONY video cameras that are developed from a single adaptable design. Design adaptability in these cameras is achieved through a shared *platform* which allows the attachment of various modules for different functions. Although these cameras are produced from an adaptable design, an individual physical camera is very difficult to modify for a new usage by the user. This example demonstrates that design adaptability primarily concerns the producer.

SONY DCRTRV350 SONY CCDTRV318

Figure 3. 4: An example of design adaptability (courtesy of SONY).

Product (User) Adaptability

The second type of adaptation, which is more directly related to the environmental benefits discussed earlier, is *product adaptability*. It refers to the ability of a single physical product to be used for different service requirements. The adaptation task is usually performed by the user, so it can be also called *user adaptability*. User adaptations include the upgrading and customization of products as well as the attainment of several functions from a single versatile product. An example of product adaptability is the development of adaptable homes, where adaptations happen continually through time as the needs or the lifestyle of the residents change (Figure 3.5). This structure allows for future additions, provides extra space for possible future use of wheelchairs, and facilitates changes in features and functions of various parts in and around the house. The adaptable design housing etiquette requires that modifications be simple to carry out and become cost effective when they are planned into the initial design of the house. The Adaptable Design Guidelines of the City of North Vancouver state: [Vancouver 2000]

> *"Adaptable Design will create livable residences for a wider range of persons than current housing design permits. Through consideration of how adaptations could be easily and inexpensively incorporated at a future time,*

Adaptable Design will allow for changes which are required by residents with varying or changing needs, thereby supporting independent living (for those with moderate disabilities)…. new developments and technology may result in equivalents that meet the intent of a specified requirement."

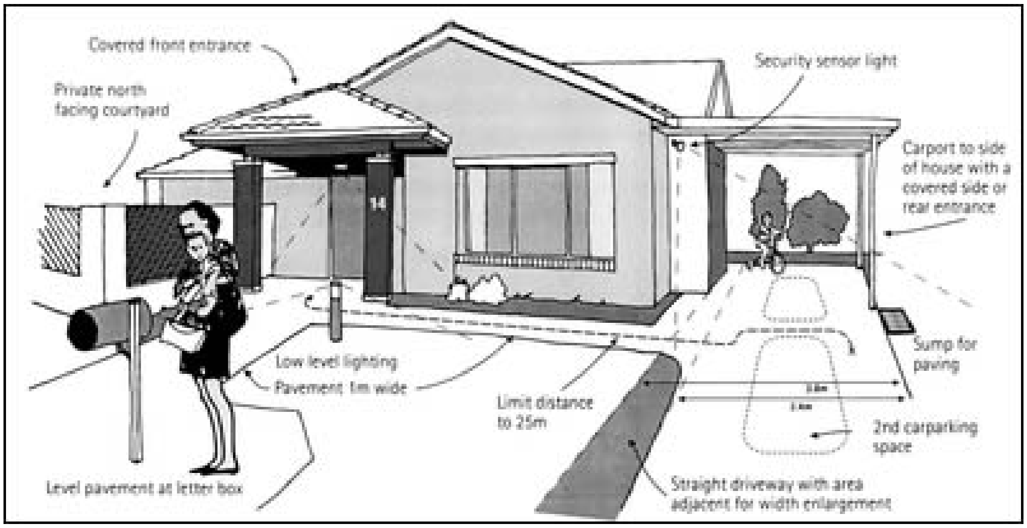


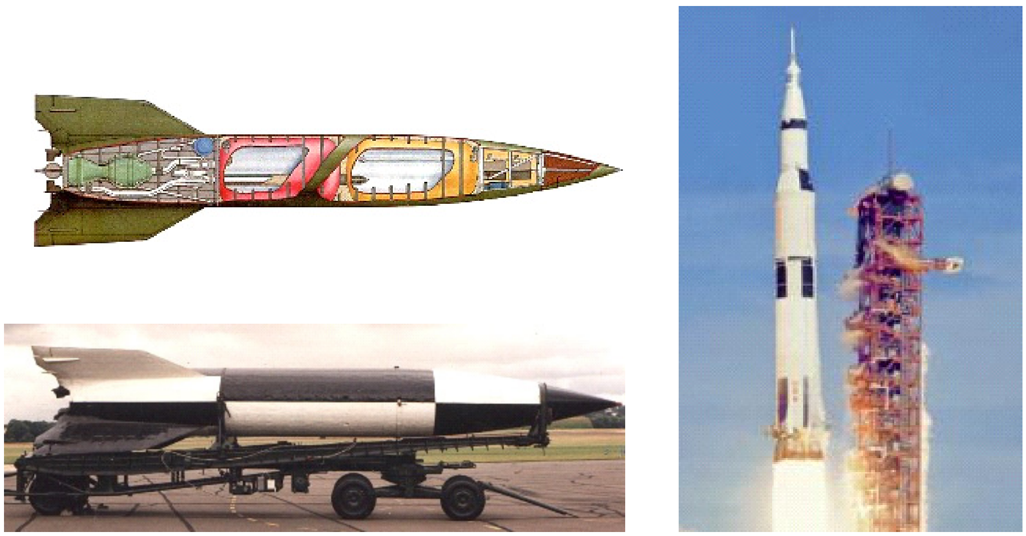
Figure 3. 5: Adaptable homes (courtesy of the City of Vancouver).

3.2.2. Sequential and Parallel Adaptations

The extension of the utility of a product or design may occur over the course of time, or it might be unrelated to time. These two categories of adaptations are called *sequential* and *parallel* respectively.

Sequential Adaptations

A sequential adaptation occurs over the course of time; it is usually the result of the emergence of new technologies or the time-related changes in service requirements. Sequential adaptability extends the 'service life' of a product or a design, and it may or may not be reversible. An example of the sequential adaptation of a *product* is seen in adaptable homes (Figure 3.5). A historical example of sequential adaptation of a *design* is shown in Figure 3.6. This figure shows the evolution of a design, where the design of the V2 ballistic missile developed by Wernher Von Braun and others in Germany during World War II was later adapted to produce the design of the American Redstone missile, which eventually evolved to the design of the Jupiter and Saturn space rockets. [Wikipedia Encyclopedia, http://en2.wikipedia.org/wiki/Wikipedia].



V2 Ballistic Missile

Saturn V Space Rocket

Figure 3. 6: An example of sequential design adaptability.

Parallel Adaptations

A parallel adaptation extends the usage of a design or a product into various applications. Parallel adaptability is unrelated to time; it extends the 'service scope' of a product or a design and the adaptation is usually reversible.

Parallel adaptability of a 'product' means that the same product can be set up in various ways to perform different functions. The adaptation of a product to various usages is performed by the user in a usually reversible and simple procedure. This typically results in the development of *versatile* products which are capable of performing several functions. In parallel adaptation of a 'design', the same design is adapted by the manufacturer to produce a variety of products for different customers, or to produce customized products at almost the same price as the standard models. The diversification of a product portfolio through the parallel adaptation of a 'design' is known as *mass customization* in the literature, as discussed in Chapter 2. Figure 3.7 shows three Ford vehicles that share not only the same general design, but also various parts and assemblies such as the chassis, headlights, engines, and accessories.

Ford Explorer Sport　　Ford Explorer Spot Trac　　Ford F150 Pick-Up

Figure 3. 7: Parallel design adaptability, performed by the manufacturer, results in variety and mass customization (Courtesy of Ford Motor Company).

3.2.3. Specific and General Adaptabilities

Unlike the conventional design process in which a product is designed for a nominal set of functions, AD develops products that can also be adapted to different or additional functions beyond their normal operational mode. Therefore, it seems that the designer must have some ideas as to what these additional requirements are, and design the product accordingly. In many cases such forecast information exists and is utilized, for example in the design of video cameras shown in Figure 3.4. This is called "*specific adaptability*" because the provisions in the design are made for specific adaptations which are known in advance. However, it is also possible to design products in such a way that they are generally more adaptable than conventional designs even if no forecast information is available. This is called “*general adaptability*”. Specific and general adaptabilities were discussed in Chapter 1. This section further discusses the four categories of specific adaptability: *versatility*, *upgrading*, *variety*, and *customization*. It should be mentioned that these categories are not mutually exclusive and there is an overlap between them.

Design for Versatility

If the additional functions which are expected from a product are definitely known during the design process, the product can be designed to deliver multiple functions including the original FRs and the additional FRs. Since in design for product versatility the additional FRs are treated the same way as the original FRs, it could be treated as a conventional design process. In this thesis, however, design for versatility is considered as a category of AD in which the maximum amount of forecast information is available. A product is designed for versatility if adaptations from one function to another occur frequently; therefore the product is designed so that these adaptations do not require significant alteration of the product and often involve a simple procedure which can be performed by the user.

Design for Upgrading

Upgrading is the adaptation of existing designs and products to new needs or technologies as they become applicable. An example of design for upgrading is the design of computer systems. An individual computer is upgradeable by the user because its rapidly-expiring components are designed as easily replaceable units. A computer is also upgradeable in its ‘design’ as the architecture of a computer is designed to allow the manufacture to incorporate new technologies.

Design for Variety

Variety refers to parallel 'design adaptability' thus it concerns the producer as discussed earlier in this section. In design for variety a single design (blueprint) is used to produce a variety of products. Design for variety is also called 'design for mass customization' or 'product family/portfolio development' in the literature. An example of design for variety was shown in Figure 3.7, where several vehicles in the company's portfolio shared similar designs.

Design for Customization

Customization is a general term which refers to the adaptation of a product or design to specific preferences. For example, various features and functionalities of SONY video cameras such as the image stabilizer, digital zoom, and the side screen are designed in the form of optional units (Figure 3.4). Then many models can be easily developed by the morphological combination of various features in response to diverse customer preferences.

Figure 3.8 shows the role of forecast information in these four categories. It can be seen that the availability of forecast information justifies more initial investment, as in versatile products, while in the absence of such information less initial investment is made because there is no certainty that adaptations will be required after the design is finished.

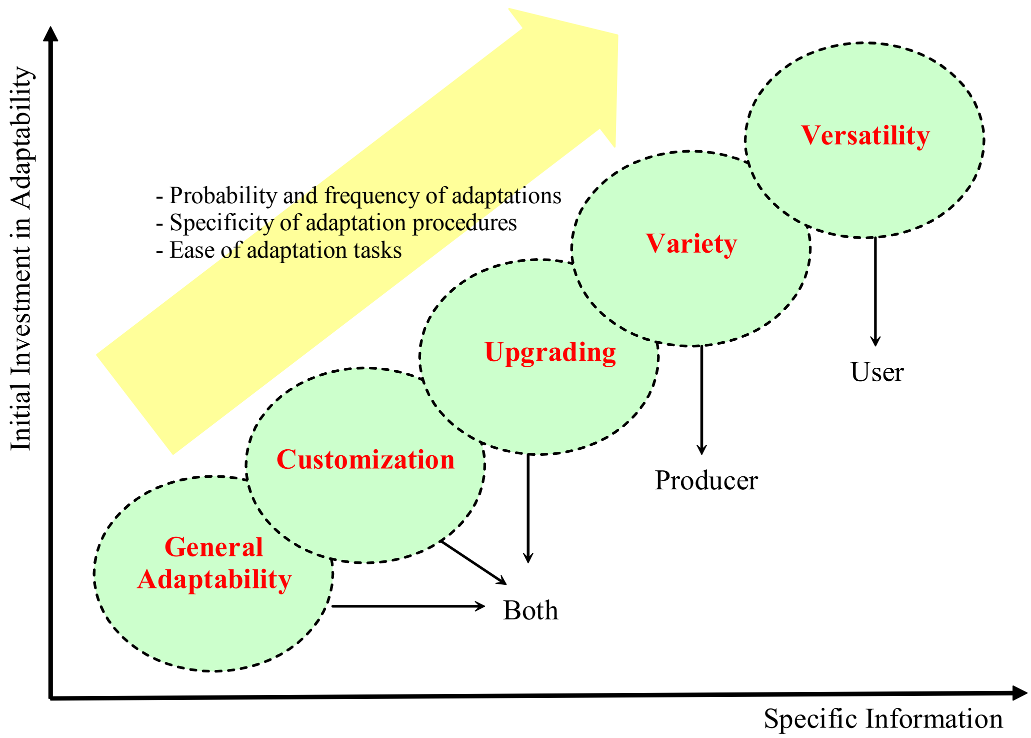


Figure 3. 8: The relations between the availability of specific information and the justification of initial investment in the four categories of specific AD.

3.2.4. Summary

According to the categorizations of this section adaptations can be *parallel* or *sequential*, and they can be applied to a *design* or to a *product*. These two divisions create four categories. The chart in Figure 3.9 shows these categories within Specific AD: *upgrading* is the sequential adaptation of both designs and products, *variety* is the parallel adaptation of a design, *versatility* is the parallel adaptation of a physical product, and *customization* is a general term for all these categories. It can be seen that such classifications are not made for *general AD* because general adaptability does not predetermine any type of adaptation.

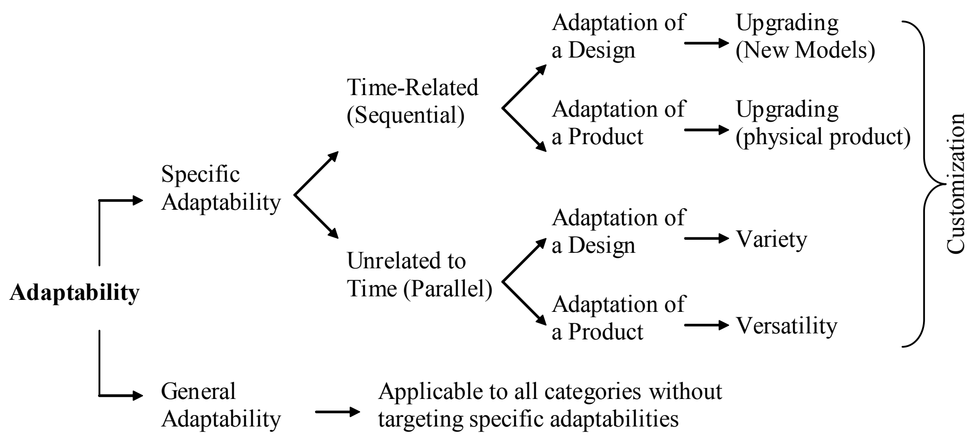


Figure 3. 9: Various categories of adaptations.

3.3. Design Categories Suitable for AD

Notwithstanding its benefits, design for adaptability also has some disadvantages. For instance designing a product for adaptability might result in additional costs and the loss of optimality in weight and performance. Therefore, it is important to decide if and to what extent AD is applicable to a design problem. The final decisions require trade-off analysis which takes into consideration the need for adaptability, the cost of adaptable design, the cost or effort of performing an adaptation task, the frequency of adaptation tasks, marketing, etc. A preliminary assessment, however, can be made based on certain characteristics of the design problem at hand. This section discusses four design characteristics which increase the applicability of AD.

Superfluousness

Superfluousness in this thesis refers to the 'unused potential' of a product and is the most important criterion in choosing a product for adaptable design. Superfluousness is a result of a product's idle time during its operational life, and more importantly, a result of its premature retirement. Products without superfluousness are those which are mainly 'consumed' by their usage. These products, such as drill bits, cutting tools, and brake pads, should be designed for durability not for adaptability.

Product Variations

An important characteristic of a design is the features it shares in common with other designs which are considered as belonging to the same family or portfolio. Such commonalities make it possible to adapt a design from one product to a similar one, thus justifying design for adaptability. Variations of a product can occur over the course of time (new models, upgrades, and customization), or they can occur in parallel as a company's product variety and diversity. Products which have a larger number of variants are generally more suitable for adaptable design. The common method of designing such products for adaptability is the use of *shared platforms* and *differentiating modules* as discussed in Chapter 2.

Environmental Impact

Some products contain materials that are hazardous or are difficult to recycle, reuse or even dispose of. Products with larger environmental impact in their life cycle yield better environmental benefits if they can be adapted.

Financial Significance

The financial significance of a design project depends on the production volume, unit price, capital investment (infrastructure), and development time. High production volume justifies more initial investment in design, including the use of adaptable design techniques to improve the fate of thousands or millions of objects after they retire. Unit price has a similar effect and it becomes increasingly important for massive one-of-a-kind projects such as a nuclear power plant. In such projects, both the physical product and the design knowledge generated should be made adaptable for future use (product adaptability and design adaptability). Capital investment and development time are important when variety is involved. That is, by replacing multiple designs with one adaptable design large capital investments and long development times will not recur for each individual design.

An Exemplary Category

Military design and production projects often have all the above characteristics. Military products typically have a long product life because they are designed for quality, durability, and reliability; yet their actual usage time is often much shorter than their potential service life. Therefore these products are *superfluous*. Military products also have many *variations* for different deployment conditions, and they have variations which evolve in the course of time. Also, some military products have high *environmental impacts* because they utilize hazardous materials in their construction or in their production processes. Moreover, military projects are often of great *financial significance* due to large investments in design, development, tests, and refinements.

These characteristics make military products a suitable category for adaptable design. Some of these products could be designed for adaptation to civil usage during their idle times or for permanent adaptation to civil usage after their obsolescence. They might also be designed for upgradeability to new technologies and new demands, so that their retirement is postponed. This is particularly beneficial when adaptation results in avoiding the disposal of environmentally hazardous materials.

3.4. Benefits of Adaptability

This section discusses the benefits of AD for the user, the producer, and the environment.

3.4.1. The User: Extended Product Utility

As discussed in Section 3.2.1, the user is mainly concerned with *product adaptability*. In this category, an adaptable product should be designed in such a way that adaptation tasks can be easily performed by the user. The user-related benefits of adaptability result from the fact that an adaptable product replaces several products in its service life, thus saving money, storage space, maintenance, installation costs, etc. It also provides the user with the possibility for customizations which are not available in the market (personalization). Figure 3.10 shows an example of a bicycle U-lock that is also a carrier rack. This versatile design enables the user to replace two products with one.

Figure 3. 10: Product adaptability, performed by the user, results in multi-purpose versatile machines. (Courtesy of Master Lock).

3.4.2. The Producer: Extended Design Utility

As discussed in Section 3.2.1, the producer is mainly concerned with *design adaptability*. The producer may use the same design, its associated process plans, manufacturing set-ups, and even existing parts and assemblies to produce different products for different clients [DeLit 2003]. Unlike 'user adaptability' in which the adaptation is performed by the customer, in this category the adaptation tasks are performed by the manufacturer in the factory, where the required tools and expertise are readily available. Therefore, the primary concern is the long-term benefits not the ease of adaptation tasks. The producer's benefits include the reuse of design knowledge, the reduction of production time and cost via intra-company standardization, the reduction in the cost of post-sale services, and gaining marketing advantage through user and environmental benefits.

3.4.3. The Environment

The existing environmental remedies for the problem of product retirement are the *recovery* techniques, which are based on redirecting the flow of used products from disposal in landfill back into the production supply chain [Hashemian 2004]. Adaptation of a product has a similar effect because it also redirects the retired product back into new service. Figure 3.11 compares adaptability and other recovery methods. In this figure, along the time axis the processes of the production supply chain occur from left to right until the product is delivered to the user. Then there is a usage phase, followed by the end point of service life (retirement) at the right end of the time axis. At this point, various recovery methods return the product to different points on the time axis.

The figure shows the cumulative environmental impact (EI) on the vertical axis. For simplicity, EI is assumed to be a linear function of production stages (represented by time in this figure). Therefore, from any point on the time axis to the end of the production chain, EI accumulates proportionally. It can be seen that the closer the returning point of a recovery method is to the final product, the less EI is created to finish the production (the area of triangles). Therefore, the priority of the recovery methods can be listed in the following order: first, durability, repair, and maintenance that extend the normal operation; second, adaptation that extends the service life in a new operational mode; third, part salvage that reuses parts as they are; fourth, material salvage that uses existing material for manufacturing new parts; fifth, material recycling that involves shredding and reprocessing raw materials.

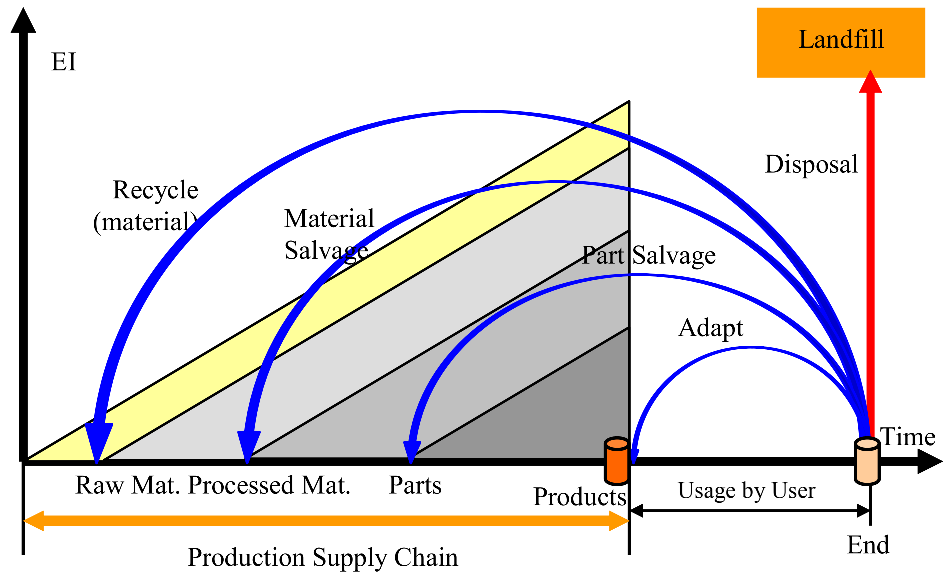


Figure 3. 11: Adaptation versus post-retirement remedies.

3.5. Summary

The discussions of this chapter are summarized in Table 3.1. The left side of the table lists various characteristics, types, and benefits of adaptability. The right side of the table describes these characteristics and benefits for design adaptability and product adaptability in two separate columns.

Characteristics of Adaptability		**Adaptable Design**	
		Design Adaptability (Producer)	Product Adaptability (User)
The one who performs the adaptation task		Producer, rarely user, by both in large projects	User, rarely producer, by both in large projects
Results of adaptation		Population of products from same design	One product, several or extended usages
Category for the extension of Utility	Sequential (in time)	Upgrading, new models, customization	Extended service life, upgrading, customization
	Parallel (in scope)	Product variety and mass customization	Versatility of products
Benefits	User	Variety of choices at lower costs, product familiarity	- One product replaces a few, various savings - Upgrading and customization - Adapting to changing needs of the user
	Producer	- Lower cost and time of design/production - Quick Market response and technology update - Market share due to variety and mass customization	- Better market share due to user benefits - Better market acceptance due to environmental consciousness
	Env.	Better use of resources during production, part salvage due to shared modules	More service with fewer products Less waste and pollution

Table 3. 1: The relationships among various aspects of Adaptable Design.

Chapter 4: Design for Adaptability

This chapter presents this thesis's method of designing products for adaptability. Section 4.1 discusses the theoretical arguments of the *function-based segmentation* approach for *general AD*. Section 4.2 discusses the difficulties of applying this approach to the design of mechanical systems. Section 4.3 presents a 'measure' for adaptability based on the concept of *information content*. Section 4.4 presents methods and guidelines which help with the adaptable design of mechanical systems. Section 4.5 describes the overall methodology of *design for adaptability* including both *specific AD* and *general AD*. Section 4.6 discusses the extension of this methodology for the inclusion of life cycle design. Section 4.7 provides a brief summary of this chapter.

4.1. Fundamentals of Design for General Adaptability

This thesis divided 'design for adaptability' into *specific AD* and *general AD*. Specific AD is an umbrella term which refers to the methods of designing a product for a predetermined set of adaptations. General AD, on the other hand, refers to designing products for general adaptability without targeting a set of predetermined adaptations.

Methods of specific AD are straightforward because their adaptability requirements are determined from the outset and these predetermined adaptations guide the design process. For example, in *design for upgrading* the rapidly expiring components of a product might be designed as replaceable modules, and in *design for variety* a family of products might be developed from a *shared platform* and several *differentiating modules*. These methods were discussed in Chapter 3.

In design for general adaptability, however, predetermined adaptations cannot be used to guide the design process because such forecast information is unavailable. Therefore general AD requires a different approach. Chapter 1 briefly mentioned that the approach of this thesis is to emulate the adaptability of a *rational functional structure* in the *physical structure* of a design. This method, which does not depend on forecast information on future adaptations, could be called 'function-based segmentation', or 'subordination of the physical structure to the rational functional structure'. This section discusses the theoretical reasoning for this approach. Although some arguments of this section are applicable to any type of design, it should be assumed that the scope of discussions is limited to mechanical engineering design.

4.1.1. The Design Hierarchy

A system may be defined by describing its functions, and without detailed specification of its mechanisms. Complex systems might be expected to be constructed in a hierarchy of levels. Then, similar to the whole system, a subsystem may be defined by describing the functions of that subsystem, and without detailed specification of its sub-mechanisms. The subsystems of an artifact at lower levels nest themselves according to a hierarchical schema in a semi-independent way, each performing the function they were designed for at the level they are nested in the hierarchy. The design of each component can then be carried out with some degree of independence from the design of others, since each component will affect the others largely through its function and independently of the details of the mechanisms that accomplish the function. This hierarchical ordering is the "*shape of design"* [Simon 1969].

Holon

A hierarchical system is not an aggregation of elementary parts, and in its functional aspects is not a network of elementary units of behavior. It is a system that is composed of interrelated semi-autonomous subsystems, each of the latter being, in turn, hierarchic in structure, until we reach some lowest level of elementary subsystems. A hierarchical structure can be represented by a multi-leveled organization of *nodes*, where each node branches into sub-nodes to form a *tree*. Each node within the hierarchic tree has two properties: it is a *whole* relative to its own constituent parts, and at the same time a *part* of the larger whole above it in the hierarchy. This dual characteristic has been called the '*Janus effect'*, named after Janus the two-faced Roman god with faces looking in opposite directions [Koestler 1967]. Any member of a hierarchy has a '*whole'* face looking down towards subordinate levels, and a '*part'* face looking up towards the apex. Not being satisfied with words such as sub-whole or sub-system, Koestler coined the term '*holon'*, from the Greek *holos* (whole) and *on* (part), to designate a node in the hierarchic tree.

Scale Independence

An important property that can be deduced from the above discussion is the scale independence of a hierarchic structure. That is, every holon in a hierarchy is bound by the Janus effect regardless of the level it is positioned in the hierarchy; this includes the apex, the middle holons, and the end-node holons. The apex of a given hierarchy, itself a holon, is a node of a larger hierarchy from which it was dissected; the end-node of a hierarchy, also a holon, can be decomposed further into its constituent elements.

The scale independency of hierarchical structures raises questions as to which holons are the apex and the end-nodes of a given hierarchy; that is, where a hierarchy begins and where it ends. For a design hierarchy, these determinations depend on the circumstances of a design task as explained below.

Apex and End Nodes

The apex of a design hierarchy is the system which is chosen for the task at hand; its parent systems are considered as the working environment of the chosen system. For instance the design of a vehicle might be the apex in a design problem, that is, 'vehicle design' initiates the task. The vehicle itself is a part of a larger system for urban transportation, involving the design of lane widths and bridge capacities. To the designer, however, the system of interest is the vehicle and road characteristics are considered as the constraints or attributes of the vehicle's working environment. As a contrasting example, the apex of another design problem might be the design of a gearbox for a vehicle; in this example the vehicle is considered as the working environment for the gearbox system.

The end-nodes are decided by a relatively arbitrary decision as to where to stop the decomposition process. In engineering design the end nodes are the unambiguous description of an artifact. These descriptions are then communicated to another entity, for whom these descriptions form the apex of a new problem. This issue is further discussed in the next section, which explains the internal mechanism by which a hierarchic tree is formed in the design process.

4.1.2. Decomposition and the Design Holon

The Design Holon

The previous section discussed three properties of a design hierarchy: the nodes in the tree structure are 'holons' bound by the Janus effect; the hierarchy is a self-similar and scale independent structure; and the apex and end holons are decided arbitrarily. The self-similarity of a design hierarchy means that it is constructed from a recurring pattern. This pattern, which is a general template for a single node or holon, is called the *design holon* in this thesis. From the Janus effect it can be deduced that a design holon should have a mechanism for producing its subordinate holons, and its own existence should be attributable to a higher holon from which it has been decomposed.

The design holon consists of three elements: a problem, a solution which is synthesized for this problem, and the decomposition of the problem into sub-problems based on the chosen solution. The reason for considering the solution as an ingredient a design holon is that, in the decomposition schema adopted in this thesis, decomposition is performed only after a solution for the problem is found. Suh has discussed the relation between choosing a solution and decomposing a problem [Suh 1990]. Describing the goals of a design task by a set of *functional requirements* (FRs), he states: "*FRs at the i^{th} level cannot be decomposed into the next level of the FR hierarchy without first going over to the physical domain and developing a solution that satisfies the i^{th} level FRs.*"

The goal of a design problem is described by a set of *functional requirements* (FRs). In order to achieve the overall goal, a solution must be devised for each FR. This process includes *synthesis*, which involves creativity and experience, and *evaluation*, which results in choosing a solution from several alternatives. Then, the realization of the chosen solution imposes its own requirements, which are the conditions that have to be met to assure the proper functioning of the chosen solution. Since these are the *requirements* for the *functioning* of the solution, they are called the *functional requirements*. These sub-FRs form new sub-holons, and this is the process of decomposition in design. This discussion also reveals that the initial goals are also called FRs because they too are the requirements of an adopted solution for a higher system. This higher system, however, is beyond the scope of the design problem at hand as discussed in the previous section.

Decomposition Rules

The decomposition of FRs into sub-FRs proceeds through "zigzagging" between the problem space and the solution space, or between FRs and their solutions [Suh 1990]. The fact that a problem (FR) needs to be solved before it can be decomposed is an important rule of decomposition adopted in this thesis.

The new FRs that are produced as the result of decomposing their parent FR should be both *necessary* and *sufficient* for the attainment of the goal of their parent FR. The sufficiency of decomposition means that when the solution for FR_i generates n new requirements, the fulfillment of these n requirements at the (i+1) level should guarantee the proper functioning of the adopted solution for FR_i. The necessity of FRs means that every FR within the set created by the decomposition process needs to be explicitly resolved by the designer in order for the chosen solution to function properly.

The process of decomposition in design changes the representation of the problem from FR_i to $FR_{(i+1)}$; which means from more uncertain and abstract FRs to more deterministic and concrete ones. The decomposition process and the subsequent functional structure are not unique. They depend on the formulation of FRs for any given node, and on the solutions chosen for each FR. Figure 4.1 depicts the elements of a design holon. It can be readily observed that this self-repeating pattern results in a hierarchy.

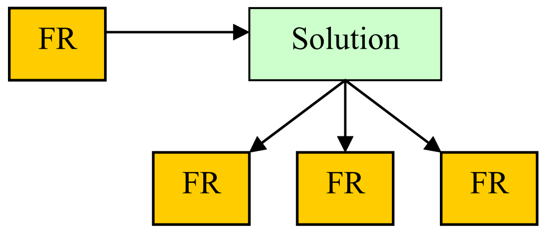


Figure 4. 1: The design holon consists of FRs, solutions, and decomposition.

Beginning from the apex, the decomposition generates new FRs in a hierarchical fashion. The question is: where does the design process end? The answer lies in reducing the uncertainty or increasing the specificity of FRs to the level of 'available' resources. In mechanical engineering, a design is decomposed to a level of specificity where FRs can be achieved by available technological resources. This is the end holon, and the designer need not be concerned with how such specific goals are fulfilled. The end holons will be further discussed when the concepts of 'task' and 'information content' are presented in Section 4.4.1.

4.1.3. The Rational Functional Structure

Using the concept of design holon, a hierarchical model of the design process can be developed. In this model, the initial design problem is represented by a set of FRs. FRs are assumed to be both necessary and sufficient for the representation of the initial goals. For each FR, solutions are synthesized and a set of solutions for FRs is chosen. The functional independence among the solutions within the set should be maintained, as suggested by Suh's *independence axiom*. Then, these solutions are decomposed into their own functional requirements. The new FRs that emanate from a decomposed FR, similar to the initial FRs of the design problem, have to be both necessary and sufficient for achieving their goal. The goal of such a set of FRs is the proper functioning of the chosen solution for their parent FR. Decomposition of FRs through this zigzagging process continues, and the process ends when the blueprint for the creation of the artifact is specified detailed enough to be sent to the manufacturer. This process results in a hierarchical structure illustrated in Figure 4.2.

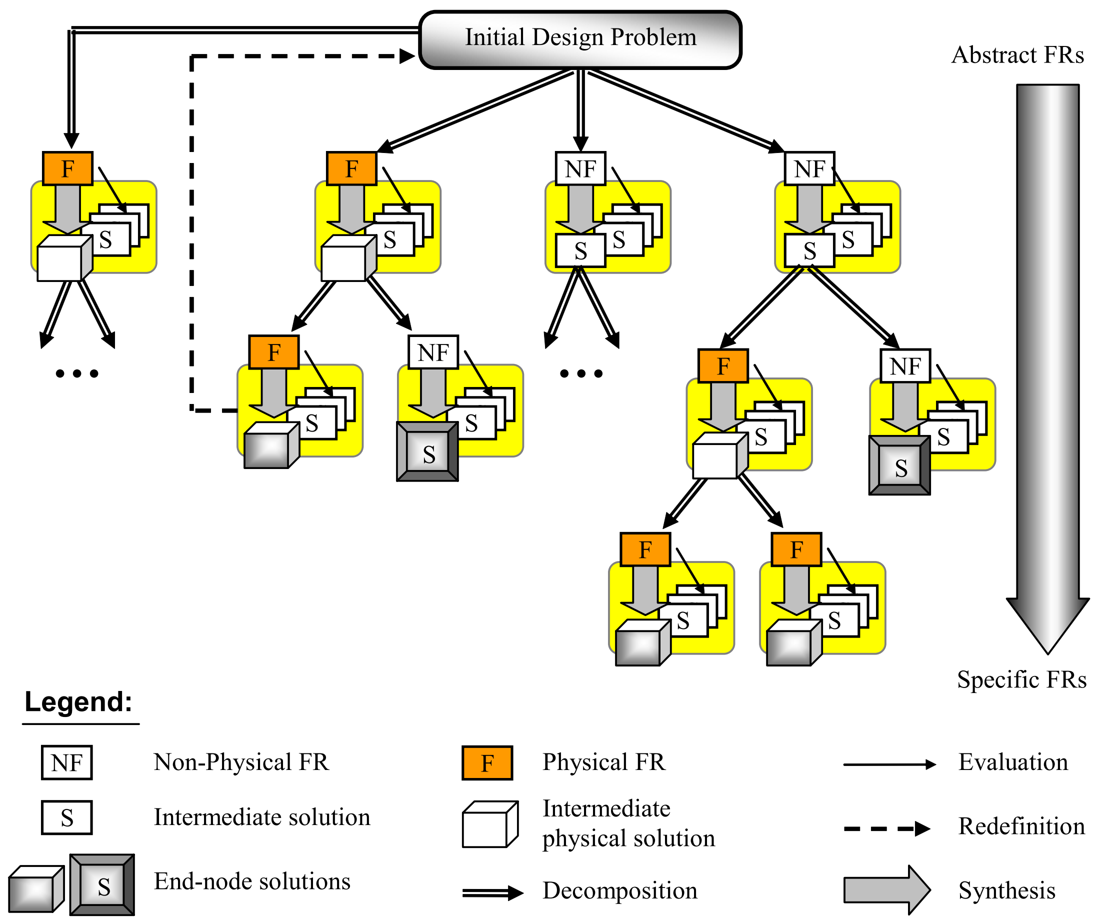


Figure 4. 2: The rational functional structure.

In the above model, the initial FRs of the design problem are divided into physical and non-physical FRs, denoted by (F) and (NF) respectively. Based on the nature of the chosen solution, a FR is decomposed into other FRs which in turn can be F or NF. The distinction between physical and non-physical FRs is important for AD because the physical structure is subordinated to the hierarchy of physical functions only.

This hierarchy represents the design rationale, and is called a 'rational functional structure' in this thesis. This hierarchy is distinguished from a more common definition of *functional structure* which is a 'descriptive' representation of functions [Pahl 1988]. A rational functional structure has certain properties which make it adaptable. These properties are discussed in the next section.

4.1.4. Causality and adaptability

In a rational functional structure described above, the relation between every node and its subordinates is a '*causal'* relation. That is, every subordinate node is a new FR which is 'caused' by the chosen solution of its parent function, and there is no other reason for this node to exist. This causality implies two properties when a node is removed from the hierarchy: first, all its subordinates become unnecessary and can be removed without affecting the rest of the structure; second, there is no need to eliminate anything else. These properties make such a structure suitable for adaptability as discussed in Chapter 1.

4.1.5. General Adaptability through Subordination

The approach of this thesis towards achieving general adaptability is to subordinate the physical structure of a product to a rational functional structure described above. This subordination results in the product having the same adaptability as its rational functional structure.

In order to achieve this subordination, the architecture of a product should be designed by the same rules which generate a rational functional structure. First, each physical function should be fulfilled by a distinct subsystem. Second, each subsystem must contain within it all the elements it needs for its proper functioning. Third, a subsystem should not have any duties other than its nominal function. In this fashion, the elimination of a subsystem does not affect the rest of the product.

The architecture of a subordinated system will be an assembly of *autonomous functional modules*. Each module, in turn, is a system which can be also developed as an assembly of independent modules. The nesting of this structure results in a hierarchy which corresponds to a rational functional structure. It might not be practical, however, to follow this division to detailed levels because small subsystems are often needed as a whole, and there is no need to develop their insignificant components as independent modules. The next section discusses the challenges of incorporating this architecture in the design of mechanical systems.

4.2. The Challenge of Mechanical Design

Few mechanical devices can be adapted to varying service requirements without considerable effort; therefore most mechanical devices stay in their normal operational mode until their retirement. The lack of adaptability in mechanical systems can be attributed to the two broad properties explained in this section.

The Nature of Mechanical Components (*Structural Connectivity*)

Adaptation is the *modification* of the internal mechanisms of a system in response to outside variations. In engineering systems, adaptation is a response to the new requirements for service or operation and invariably involves the modification of the internal structure of the artifact. For example, the addition of a room to a house is the internal modification that adapts the house to the new spatial requirements. Therefore, the *adaptability* of a system is a reflection of its *flexibility in allowing the necessary internal modifications*.

Performing structural modifications is particularly difficult in mechanical systems. The reason is that the functions of mechanical components are achieved via their forms and the geometrical or spatial orders among interacting components. Any modification in the structure of a component, difficult and costly by itself, may also disturb the spatial and geometrical relationships, and thus affect the function of several other components. These components, which may be functionally independent from a logical point of view, are often connected via various constraints such as size, shape, alignment, adjacency, attachment, closure, motion, direction, etc. As a result, any modification is often propagated throughout the product, hence making the adaptation process costly. This property can be called *structural connectivity*.

Since structural connectivity is an inherent property of mechanical systems, the available remedies for reducing propagation of changes are limited to two categories: *use of alternative technologies*, and *segmentation of the structure*. The first category basically avoids the use of solid components and their associated spatial constraints, and replaces them with hydraulic, electronic and software systems. The second category is based on the premise that in a modular structure modifications are likely to be confined within segments and are less prone to propagating into other segments. Modular design, platform design, interface and bus system design, and manufacturing adjustment design are examples of design approaches for the segmentation of mechanical systems ([Otto 1994], [Lee 2003], [Chen 1994], [Yu 2003], [Gu 1997], [Gu 2003]).

The Nature of the Mechanical Design Process (Functional Ambiguity)

An engineering product is expected to deliver some *physical functions* related to materials, energies, and signals; the design process involves identifying the required functions and finding solutions for them ([Pahl 1988], [Suh 1990]). The relation between a function and its solution is a *causal* relationship, which means that a solution would not exist if its function was not required. Ideally, each function is performed by its corresponding solution independent from other components in the product.

If the solution for a function is a complex system, this solution may impose its own functional requirements for which the designer must find solutions as well. This is the process of *decomposition* by which the design proceeds from abstract and complex functions to more specific and simpler functions. At every level of decomposition, the relation between functions and their solutions remain causal, as described above. Section 4.1 discussed an ideal rational functional structure, which is a hierarchy of causal relationships where every function in the hierarchy is only related to its parent function and to its subfunctions. This section also discussed the adaptability of such a structure: if a function is no longer required, it can be eliminated together with all its subfunctions and their corresponding solutions, and there is no need to eliminate anything else.

Unfortunately, the conventional mechanical design process does not result in an ideal scenario, and the hierarchy of physical assemblies does not reflect the rational functional structure for two reasons.

First, a mechanical design typically follows the ideal hierarchy for only one level of decomposition at a time. If at any point of the design process the number of required functions is n, the designer needs to find n solutions for these functions. There is a causal relationship at this level and every solution exists because of its corresponding function. Then, some of these solutions may impose their own requirements. The new requirements for a solution can be caused by both the functional elements of the chosen solution, and by the *constraints* that relate this solution to other solutions. The latter may disturb the hierarchical tree of functions because a newly emerged function can be listed under several higher functions in the hierarchy not just under its parent function. As the design proceeds, various constraints become increasingly important in driving the design process, and it becomes unclear how some new subfunctions are related to the original design problem [Whitney 1993]. Therefore, if an initial design function is eliminated or the solution for it changes, as is the case in adaptation, it is not always obvious which components ought to be removed. For example, the grill in the front of a car, a part of exterior design, is functionally related to the internal combustion engine. If the engine is replaced by an electric or hybrid engine, there will be no functional need for the grill.

Second, due to omnipresent constraints on weight, size, and cost, *component sharing* is an integral part of the mechanical design process. Therefore, several functions at various levels of the design hierarchy can be achieved via several design features (*design parameter* or DPs according to Suh) of a single component or assembly. This component sharing is often necessary because part redundancy in mechanical systems is usually unaffordable. For example, the main function of an engine of a vehicle is to provide power to the transmission system; but it also drives the alternator and water pump, provides crash safety for passengers, and provides heat for the cabin in winter.

For these reasons there is often some ambiguity as to the function of mechanical components. A component may contribute to several functions, yet it may not be autonomous in fulfilling a single obvious function. This makes mechanical components *product-specific*, as opposed to *function-specific*. This is the reason mechanical parts are difficult to use in any other product than the original product they were designed for. Current modular design techniques provide little help in this regard because their clustering algorithms are based on physical parts not based on functions [Gu 1999].

Mechanical Design and Other Engineering Designs

Figure 4.3 shows the difficulties of adapting mechanical systems in comparison with some other engineering systems. Software systems are the most adaptable category of engineering systems. The connectivity among various parts of a software system is limited to the exchange of input and output variables, and the modularization of software is always based on functions. Electrical systems involve physical embodiments and constraints; thus they tend to have some structural connectivity and may be product-specific. Buildings and other structures related to civil engineering have more structural connectivity than electrical systems, but their functions are generally clear, and a civil engineering system can often be used in different designs and configurations. Mechanical systems are the least adaptable, because mechanical design imposes both structural connectivity and functional ambiguity.

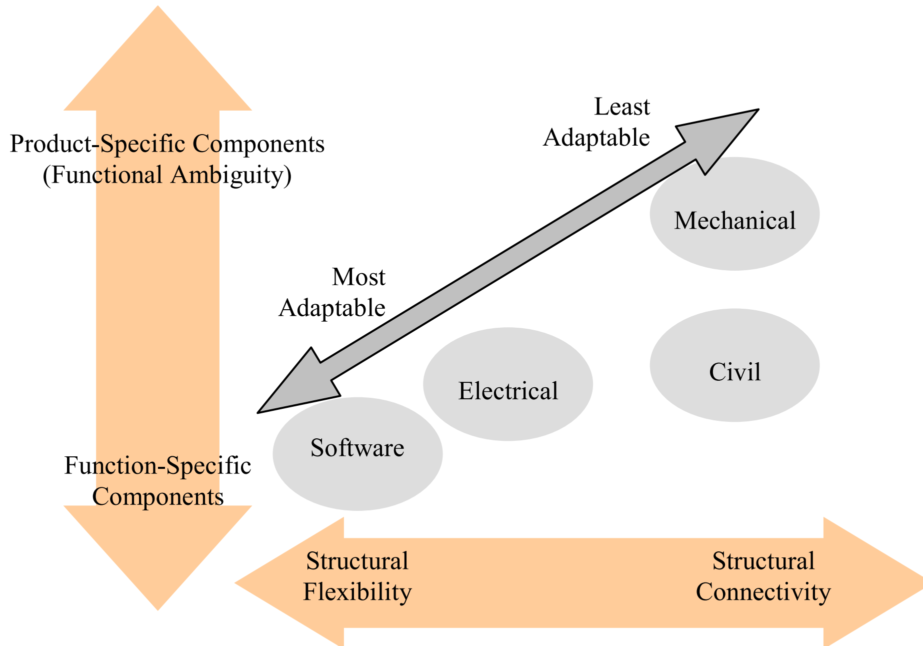


Figure 4. 3: Mechanical systems are generally less adaptable than other engineering systems.

Discussion

A subsystem within a mechanical product may contribute to several functions while it might not be sufficient for a single useful function because this function is obtained from the work of several subsystems. As a result, the function of a mechanical subsystem might be ambiguous and this subsystem might be usable only the context of its original product. The physical attributes of a subsystem, as well as the design requirements of its mating parts and interfaces with other assemblies, often make the subsystem even more product-specific. Also, due to the presence of geometrical and other physical constraints, the functional independence between two design parameters does not mean that they are structurally independent or that they are even separate parts. The example of a can/bottle opener ([Suh 1990], p51) satisfies the independence axiom, but the two FRs (open can, open bottle) are obtained from two independent design parameters that are structurally implemented on a single part. Further, the decomposition of an FR into new sub-FRs in mechanical design is driven by both *functions* and *constraints*; after a few levels of decomposition there will be many FRs in the functional structure that have no obvious relation with the original design problem.

Due to these properties, the physical structure of a mechanical system typically does not correspond to a rational functional structure (Figure 4.4). In such a case as the one depicted in the figure, when a function is removed from the functional structure on the left, it is difficult to identify and implement the necessary changes in the physical structure on the right. This chapter will present methods and guidelines which help in subordinating the physical structure of a mechanical system to the rational functional structure as much as possible thus enhancing the general adaptability of designs.

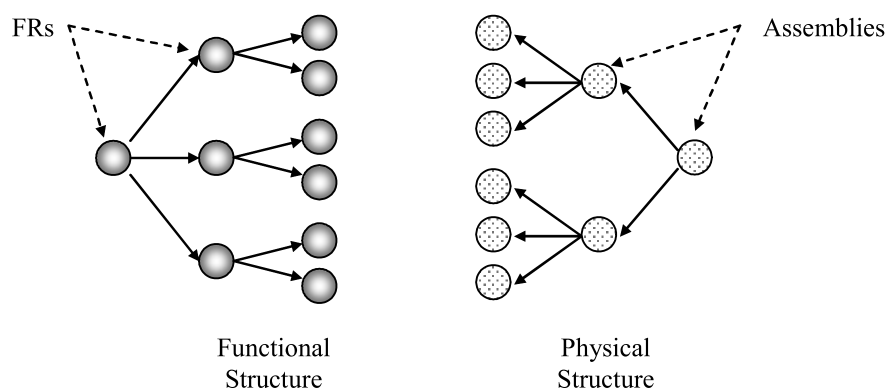


Figure 4. 4: Functional and physical structures may not correspond.

4.3. Measure of Adaptability

The term 'adaptability' for a product refers to the ability of the physical product to be adapted to a new operational mode; for a design (blueprint) it refers to the ability of the design to be adapted to produce a new design whose physical manifestation can serve in a new operational mode. In the first case, the subject of adaptation is a physical product, and in the second case, it is the design of a product. These are called 'product adaptability' and 'design adaptability' respectively, as explained in Chapter 3. This section discusses the quantification of adaptability for products only; and the discussions are not repeated for design adaptability. Unless the distinction between the design and product adaptabilities is explicitly mentioned, the statements can be generalized to include both.

An *'adaptation task'* is the actual process of modifying a product from its current state to a new state that enables the product to function in the new operational mode. A quantitative measure of adaptability should be an indication of how easily an adaptation task can be carried out. This depends on both the design of the product and the adaptation task in hand. Therefore, in measuring the adaptability of a product, the adaptation tasks that are the target of the measurement have to be specified. This schema for measuring adaptability requires a criterion for quantifying the ease or difficulty of performing an individual task. This section first develops this criterion called the '*information content'*; then it presents the formula for measuring the adaptability of a product for a target set of adaptation tasks. The implications of this formula lead to the development of useful guidelines for AD, which are discussed at the end of this section.

4.3.1. The Information Content

The fundamental element of design is the presence of *goals*. There are always *limits* on the ways a goal can be achieved. Therefore, the attainment of goals requires a plausible plan, which is called a *design*. A design is plausible if its execution delivers the goals. The execution of a plan involves the consumption of time and natural/artificial resources. The resources that are consumed in order to achieve goals may also be called *costs*. In this section, the *information content of a design* will be defined as *the total costs of achieving goals* by that design.

Goals

While *natural sciences* study the existing laws and phenomena of nature, *design sciences* are related to *human intent* and the attainment of *goals*. The latter has been called "*the sciences of the artificial*" by Herbert Simon [Simon 1969]. Simon states that the term artificial refers to man-made artifacts, and that the distinguishing feature of man-made artifacts is that they are created to serve *human purpose*.

Goals may be described by a desired state of things. That is, in an artificial activity the goal is to change an existing or naturally-occurring state of the environment to a preferred state. The alteration of the existing state requires an artifact, without which the preferred state would not occur. The artifact requires a design. Therefore, the goal of a design task is to create an artifact that achieves the preferred state. If the desired state could be achieved without an artifact, there could be no talk of design. In engineering, the design task is to generate the blueprint or instructions for the creation of an artifact that performs a set of predetermined functions, thus achieving the desired state in the working environment of the artifact.

Limits and Constraints

The working environment of an artifact includes both natural and artificial objects. The behavior of the environment is bound at all times by *laws* and properties that govern its dynamic states. For example: physical laws govern the trajectory of a projectile and the flow of thermal energy between objects; government regulations may affect the price and availability of certain materials at a given time; market trends may affect the service life of a design; and limitations in human physical abilities may constrain the operation of a device.

Further, the limitedness of resources of all kinds may impose various constraints on a plausible solution. For instance, the limits on the steady-state pollution absorption capacity of a river might constrain the acceptable designs of a sewage processing plant; the limits on the amount of available solar energy might constrain the design of solar vehicles; the limits on time might constrain a project's plan; and so on.

As a general rule, these laws, properties, and limitations exist as facts and cannot be controlled by the designer. In a design problem, these limitations are usually interpreted as '*constraints'*. Some constraints are explicitly mentioned in a design task, and some are considered as general knowledge. Due to these constraints, changing an existing situation to the target situation requires *work* or *effort*. In engineering design, these constraints are responsible for the costs and complexities of creating an artifact that satisfies the FRs.

Task and Information Processing

In this section, a *task* refers to a process that begins with a set of goals and ends when these goals are achieved. Setting the goals is not within the scope of a task itself. Goals represent the desired state, which does not exist or occur automatically through the existing trend of the environment. The goal of an '*engineering task*' is to create an artifact with the desired functions and specifications. Achieving this goal involves design and production, and requires *work*. This thesis uses the term *information* *processing* (IP) to refer to the work or effort spent to accomplish a task. This term indicates that every task ultimately consists of making decisions and allocating limited resources (Appendix 3).

There is no single common means to measure, compare and trade various types of IP capacities. For instance, a CNC machine simplifies a task and reduces the amount of IP required. However, this machine utilizes more complex technology, and requires a skilled operator. These are scarce resources and require more IP to obtain than a simple machine with a less skilled operator. Although it is difficult to effectively quantify IP, monetary value often proves to be an efficient method to measure IP and trade it between different service providers.

The Information Content of a Design

A task is always accomplished according to a plan or a course of action, which can be called its '*design*'. Simon stated: "*Everyone designs who devises courses of action aimed at changing existing situations into preferred ones*" [Simon 1969]. A design determines the path from the statement of goals to the fulfillment of those goals. The amount of work needed to finish a task depends on this path. The *information content* (IC) of a design is the total IP needed to accomplish the task (attain goals) by following that design.

In engineering, a task is the creation of an artifact that is capable of delivering the required function. A design is *finished* when the blueprint for the creation of the artifact is specified to a level that can be communicated and contracted to available service providers, with reasonable certainty that they will successfully achieve their task. This is the end node in the hierarchical decomposition process. The *IC of a finished design* is measured by the total costs of realizing it. The total costs of a design represent the amount of IP required to attain the goal according to that design, which is the total cost of manifesting the design in the form of an artifact which successfully delivers the required functions.

The *IC of an incomplete design* is more difficult to determine. Incomplete designs exist at every middle node in the design hierarchy. These designs are the different alternatives that the designer has to choose from during the design process. It is difficult to determine if a particular choice will result in a successful design at the end of the design process. If a design proves to be unsuccessful, it should be regarded as *iteration* in the design process. The IC of an unsuccessful design can be considered '*infinity'* because such a design does not lead to the accomplishment of goals and one can view it a requiring infinite amount of work to succeed. If an incomplete design yields a successful design when it is finalized, its IC is the sum of the IP of finishing the design task and the IC of the final design. In other words, the IC of an incomplete design is the cost of finishing the design task plus the cost of materializing the finished design.

Summary and Conclusions

- The *Sciences of the Artificial* are about achieving *goals*; a goal is a *preferred state*; changing the existing state to a preferred state requires an *artifact*; and the creation of an artifact requires a *design*.
- Due to the presence of constraints, any alteration in the state of environment requires *work*, which can be translated into *information processing*. The amount of this IP depends on the design of the artifact which is intended to achieve the desired state, and is called the *information content* of that design.

- The IC of a design can never be zero. Zero IC means no change of states, hence no design.
- The IC of an ideal design approaches zero. This is consistent with Altshuller's law discussed in Section 2.1.2.
- The best design is the closest one to the ideal design.

The rest of this section develops a formula for measuring the adaptability of products based on the notion of IC developed above. The following symbols are used:

Symbols

P	A single product
Sp	Set of relevant adaptation tasks for P
Spi	A single adaptation task for product P
Pr (Spi)	Probability or frequency of Spi
Inf (Spi)	The information content of Spi
A (P)	Adaptability of product P for the set Sp
AS2	Actual Final State
IS2	Ideal Final State
IMIC	Ideal Minimum Information Content
AF	Normalized Adaptability Factor

4.3.2. General Measure of Adaptability of a Product

Assume a product ***P*** that has to be adapted to fulfill some new requirements that are not delivered by its current operation. This task is an "a*daptation task*" for P. The set of "relevant" or "target" adaptation tasks for P is denoted by ***Sp***. Sp is always a sub-set of an indefinite set, which cannot be exhaustively known, of all plausible or potential adaptations for P in its lifetime. Sp is usually chosen by higher management decisions based on the circumstances that determine which adaptations are relevant to assessing the adaptability of P, especially in comparison with other products.

A single adaptation task for P is a member of Sp and is denoted by ***Spi***. Accomplishing Spi requires a certain amount of work, which represents the level of complexity of achieving the task amidst constraints and with limited resources. As discussed earlier, this is called the information content of Spi, denoted by ***Inf (Spi)***, and represents the total costs associated with performing Spi including time, resources, financial costs, etc.

The members of Sp may not be of equal importance; some adaptation tasks are more likely to occur or are more frequently needed than others. The probability of the occurrence of a single adaptation Spi is denoted by ***Pr (Spi)***. The higher the Pr (Spi), the more important it is to make the product adaptable for Spi. For example, the replacement of drill bits is an adaptability that is certainly needed in a drilling machine; therefore all drilling machines have a mechanism to facilitate this process.

In this thesis, *Adaptability* is a measure of the suitability a product for adaptation to varying service requirements. Adaptability for product P is denoted by ***A (P)*** and is calculated over Sp, which is the set of relevant adaptations. Adaptability is adversely proportional to the total cost of performing adaptation tasks within Sp. For a given P and Sp, A(P) and can be written as:

$$A(P) \propto \frac{1}{\sum_i \Pr(Sp_i) Inf(Sp_i)} \tag{1}$$

4.3.3. Physical States and IC of Adaptation

Inf (Spi) depends on how much the design of product P facilitates the task Spi. For example assume that P is an electrical hand drill to be adapted for a grinding task. The IC of this task is very small because the machine is already designed to perform this adaptation by simply replacing the drill bit with a grinding wheel through an easily detachable interface such as a chuck. As another example, assume that P is a car, to be adapted to carry large loads. This adaptation requires major modification of the car to transform its passenger cabin into cargo space. The IC to carry out this adaptation is considerable unless the car is designed for this adaptation. For example, the car may be designed so that the passenger cabin is mounted on the platform of a cargo space and can be easily removed. It can be seen that the IC of an adaptation task depends on how much the product has to be modified in order to deliver the new requirements.

Assume that the current physical state of product is S1. If at this current state the product can fulfill the new requirements demanded by Spi, the information content for the adaptation task is zero because no modification is required. This situation typically happens only if the product is designed for versatility and can perform several functions, often requiring little or zero work to switch from one function to another. Generally speaking, however, the fulfillment of the new requirements demands a change in the state of the product to a new state, or S2. Various changes in the physical states of P can be chosen to make the product capable of delivering the new requirements. S2 represents the option which is chosen, assumed to be a logical choice that involves the minimum information processing required with the starting point being S1. It is assumed that once the product is put in the final state S2, it can successfully deliver the new requirements of the adaptation task Spi with no further information processing. Thus, the information content of Spi depends on the work needed to change the current state of the product (S1) to the required state (S2):

$$Inf(Sp_i) = F(S1, S2) \tag{2}$$

Actual and Ideal States

Spi demands the product fulfill a set of requirements different from the requirements that it is currently fulfilling. If a new machine was designed, the state of this machine would involve only the necessary features and functions. We call this the "*Ideal State*", in the sense that it is the physical state which is the "minimal" embodiment needed to satisfy the new requirements. The ideal state is similar to the TRIZ concept of *"S-Field"* discussed in Chapter 2. During an adaptation task, the creation of the ideal state may interfere with the existing state of the product and require extra work. The final state of the product, after all the necessary changes have been made and the product is capable of delivering the new requirements, is called the "*Actual State*". This state has to provide for both the ideal state (minimal required embodiment) and all the necessary changes in the state that are required to assure the machine operates properly (Figure 4.5). The actual final state after the adaptation process is denoted by ***AS2***. Thus equation (2) is modified as follows:

$$Inf(Sp_i) = F(S1, AS2) \tag{3}$$

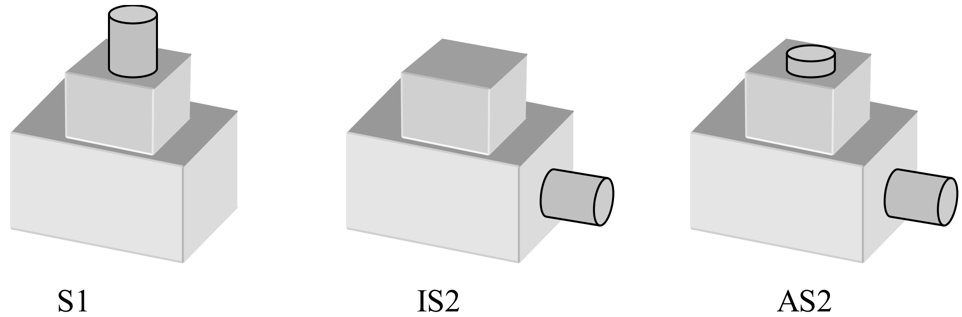


Figure 4. 5: The Ideal and Actual States.

The "Ideal Minimum" Information Content of Adaptation

If we denote the information processing required to change the product from one state to another by $Inf_{(Si \to Sj)}$ we can write:

$$Inf_{(S1 \to AS2)} \geq Inf_{(S1 \to IS2)} \tag{4}$$

The right side of the equation is the "Ideal Minimum Information Content" (IMIC) for the adaptation of P to the new requirements imposed by Spi. IMIC is the cost of creating the minimal physical state that P needs in order to deliver the new requirements. This involves only adding the missing elements and components to the current state of P. The actual IC of adaptation is higher, however, due to propagation of modifications beyond the minimal required physical state. The work required to accommodate the propagated changes is the difference between the ideal and actual costs of adaptation. The sum of such extra costs and IMIC is the actual IC of adaptation.

The distinction between IMIC and the extra work is important because the former is related to what is available in P, while the latter depends on how P is designed and constructed. The IMIC can be lower if the product, before adaptation, possesses some of the required features of the final state IS2. This can be provided for in the initial creation of P if Spi is foreseen; this is design for versatility discussed in Chapter 3. The extra work included in the actual IC of Spi can be reduced if the product's design does not propagate changes, for example if it has a modular architecture that confines modifications within one segment.

As an example consider a car which is being adapted to perform the function of a pick-up truck. The IS2 in this case needs the missing cargo box, which consists of a platform, enclosure, and an access gate (Figure 4.6). This structure is not available in the product, the car in this case, and needs to be added. Other functions of the vehicle such as mobility and control are already available.

Figure 4. 6: IS2 for the truck example.

Extra work is needed to implement the manifestation of IS2, such as removal of seats, reduction of cabin space, and replacement of struts (Figure 4.7). Such changes are necessary because of the way the car is designed; they do not directly help with the achievement of the function, which is achieved by the cargo box.

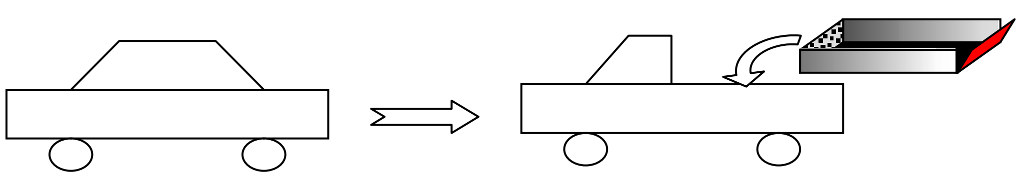


Figure 4. 7: AS2 for the truck example.

As another example, assume a design requirement that is to transfer water from a river to a reservoir. The chosen solution is a pump, and a car is to be adapted for this task because it possesses the mechanical energy needed to drive the pump (Figure 4.8). Therefore:

P: A car

Spi: Transfer water

IS2: A rotary pump

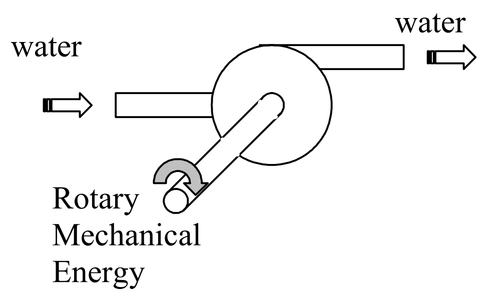


Figure 4. 8: IMIC for the pump example.

By looking at the functional structure of a car, it is easily seen that the rotary mechanical energy is available. Therefore all the prerequisites for it (such as engine) already exist in the physical state of a car. Therefore, the IMIC only requires the inclusion of the pump. Any further work required to do this is the difference between Inf (S1, AS2) and Inf (S1, IS2). This difference indicates how well the original design of the car suits this adaptation task. For example if the vehicle is equipped with a utility output shaft, a pump can be easily attached to it.

4.3.4. Calculation of Adaptability

Equation (1) indicated that general adaptability of P is inversely proportional to the IC of the members of the set Sp. However, the information content of adaptation cannot directly appear in the equation in a linear proportion. That is, if the IC of a task doubles, it does not mean that adaptability is half. In fact, the element that is directly proportional to the general adaptability of a product is the amount of '*saving'* that is achieved by adapting the product as opposed to manufacturing a new product. This measure is also more intuitive because if this saving is zero, adaptability is zero and if this saving doubles, we can say that the product is doubly adaptable.

The amount of savings achieved in an adaptation task can be represented by ($Inf_{(ZERO \to IS2)} - Inf_{(S1 \to IS2)}$). This is basically the total cost, or IC, of making a new machine minus the cost of adaptation. If this value is negative or zero, then there will be no saving by adaptation and the procurement of a new machine may have to be considered. This saving can be divided by the cost of building the new machine in order to "normalize" this parameter. Thus, for a given adaptation task Spi we define a parameter called the "*Adaptability Factor*" (AF). The adaptability factor for a single adaptation task Spi is an indication of normalized saving achieved by Spi.

$$AF(Spi) = \frac{Inf_{(ZERO \to IS2)} - Inf_{(S1 \to AS2)}}{Inf_{(ZERO \to IS2)}} = 1 - \frac{Inf_{(S1 \to AS2)}}{Inf_{(ZERO \to IS2)}} \qquad (5)$$

We can readily see that:

$$0 \leq AF(Spi) \leq 1 \qquad (6)$$

Equation 6 indicates the boundary values, 0 and 1, for the adaptability factor. If AF for a task is zero or negative, adaptation does not apply. Also, A.F. cannot be more than one because AF=1 means that the adaptation task has zero information content. The AF of one occurs in versatile machines which, from the beginning, are designed to be capable of delivering the extra service.

The goal of AD is to design-in adaptability for a product during the initial conceptual design of that product. In design for ‘specific adaptability’, target adaptation tasks are known in advance. Therefore, an adaptable design should provide solutions for some additional FRs, which are needed for future adaptation tasks, in addition to solutions for the initial FRs of the conventional design. Therefore, the total amount of savings is the difference between the *IC of multiple products* and the *IC of a single adaptable product*. Therefore, a general measure of the adaptability of a product can be obtained through the combination of Equations (1) and (6):

$$A(P) = \sum_{i=1}^{n} \Pr(Spi) AF(Spi) \qquad (7)$$

And by substituting the adaptability factor AF(Spi) from Equation 5 we have:

$$A(P) = \sum_{i=1}^{n} \Pr(Spi)(1 - \frac{Inf_{(S1 \to AS2)}}{Inf_{(ZERO \to IS2)}})_i \qquad (8)$$

Because every machine performs at least the initial set of requirements for which it was designed, no adaptation for that task is required and both probability and AF are one. Therefore, in case a machine does not adapt to any function other than its intended function, its adaptability is one. Thus we can write:

$$A(P) \geq 1 \qquad (9)$$

Depending on the value of *n* in Equation 8, the adaptability of products can range from one to five or ten or more. The adaptability of a machine, measured by the proposed scheme, gives an estimate of the number of machines that can be replaced by a single machine. For example if a machine performs three separate functions, its adaptability would be three.

4.3.5. Implications of the Adaptability Equation

- If $Inf_{(ZERO \to S1)} \geq \sum_{i=1}^{n} Inf_{(ZERO \to IS2i)}$ then this *adaptable design* is not justified.
- If $Inf_{(S1 \to AS2)} \geq Inf_{(Zero \to IS2)}$ then the *adaptation task* Spi is not justified.
- For a given product P and given adaptation task Spi, the amount of IMIC is fixed (with the legitimate assumption that IS2 is decided). IMIC can only be reduced at the initial design and construction stages of P. An initial design that has features of IS2 or includes add-on modules, however, adds to the initial costs and can be justified only if Pr(Spi) is high.
- The IC of extra work, caused by the implementation of AS2, can be reduced if modifications are not propagated throughout the product. A modular architecture with flexible and standard interfaces can confine modifications.
- Access to both utilized and potential functions of P must be increased through the subordination of physical assemblies to meaningful and recurring functions and the design of assemblies that have accessible and generic input and output connections for energy, material, and signal. The development of assemblies as functional modules is likely to increase the number of useful adaptations, which is ‘n’ in the equation.
- The cost of adaptability is the sum of the IC of developing an adaptable design and the IC of the adaptation task. This cost should be less than the IC of creating

two separate products in order to justify AD. Adaptability thus is more applicable when resources are scarce and a reduction in their consumption justifies the higher costs of AD.

4.4. Methods and Guidelines

As discussed in Section 4.1, the design hierarchy is developed by a recurring pattern of nested divisions. The careful planning of the hierarchy, which is the key to achieving adaptability, can be for the most part guided by the forecast information about possible or expected adaptations if such information is available at the time of design. In the absence of such information the general adaptability of a design to unforeseen circumstances can be improved if certain etiquettes are observed.

This section first describes the criteria for the development of functional structures for specific AD. Then it presents guidelines for the development of generally adaptable mechanical products through the subordination of a design to a rational functional structure as discussed in Section 4.1. The division of guidelines for specific and general adaptabilities is not critical as all these guidelines should be considered in the overall methodology of *design for adaptability*, which will be discussed in Section 4.5.

4.4.1. Specific AD

The development of the functional structure begins with determining the initial FRs. This process is not unique; the designer may establish the FRs in various ways. Also, the decomposition of FRs is not unique and depends on the designer's choices of solutions for every FR, and the way he/she decomposes every FR according to its adopted solution. Therefore, especially early in the design process, there is some degree of freedom in determining the functional structure, which subsequently influences the corresponding physical structure. Careful development of the functional structure results in the development of physical assemblies that perform meaningful or recurring functions. Physical assemblies developed in this way can be used in different products or configurations, hence facilitating adaptations. In specific AD the available information on future adaptations can be utilized to develop a functional structure which is suitable for foreseen adaptations. Specific adaptations in Chapter 3 were divided into overlapping categories: *versatility*, *variety*, *upgrading*, and *customization*.

Product versatility, primarily relevant to the user, was defined as the ability of a single product to be reconfigured for various functions. A versatile product may replace multiple products, and its creation is justified when all these multiple functions are needed in the normal service of the product. The additional functional requirements (AFR) can be incorporated into the initial FRs. In this case the functional structure for various FRs is developed in such a way as to take maximum advantage of the existing functions and features of the product.

Product variety, primarily relevant to the producer, was defined as using the same basic design and production infrastructure for the development of various models. In this case the functional structure for every model consists of a *base platform* and model-specific *differentiating modules*. The methods of detecting commonalities and optimizing the compromise between differentiation and commonalities were discussed in Chapter 2.

Upgradeability is similar to variety with the added factor of time. Upgrading means the utilization of new technologies or capabilities to enhance existing products for the user, or to enhance existing designs for the new production by the producer. The main technique, similar to that of variety, is modularization where frequently-upgraded parts are designed as easily replaceable modules. Customizability is a broader category. It applies to both product (user) and design (producer) adaptability. It can also be unrelated to time similar to variety, or sequential in time similar to upgrading. The basic technique in this category is also modularization.

From this discussion it is clear that depending on the objectives of specific AD, different methods are used for the development of the functional structure of a product: versatility involves multiple functions, variety involves component sharing, and upgrading involves component replacement. These methods have been discussed in the literature and were reviewed in Chapter 2. The following describes some guidelines related to specific AD.

- Define the primary FRs and the additional FRs (AFRs) for versatile product design. Find low-cost solutions for achieving AFRs in the original design.
- Provide extra features and functionalities in a design for possible future needs. For example, an output utility shaft can be (and usually is) provided in the design of farm tractors; thus making the tractor adaptable to various functions.
- Utilize the *existing* features and components to achieve extra functionalities. This guideline is against the rules of general AD which encourage the fulfillment of functions via separate subsystems. For example, an axe handle can be designed to have mounting contours on ‘both’ ends, thus enabling the user to extend the usage when one end breaks.
- Identify a group of products that can be developed from a shared adaptable

design. Identify common or recurring elements, either functional or structural, among products within the portfolio. Design these common elements as a shared platform.

- Identify the differentiating features among products within a portfolio and design them as add-on modules. Develop a parametric design for these modules so that they can be custom-made for various specifications.
- Design the interface between platforms and modules for easy attachments and detachments, such as self-aligning and lock-and-release mechanisms.
- Facilitate the replacement of components which are likely to require upgrading, such as the components that undergo rapid technological obsolescence.
- Identify customizable features, often found at the output ports where the functions of a product are delivered to the outer environment. Then design a product for the easy alteration, replacement, or addition of these features.

4.4.2. General AD

Chapter 1 and Section 4.l of this chapter explained the approach of this thesis for achieving general adaptability in mechanical systems. This approach is based on the principle of *segmentation* discussed in Chapter 1, and it was suggested that in the absence of forecast information *functions* can be used as the segmentation criterion, hence calling the approach *function-based segmentation*. Section 4.1 discussed the adaptability properties of a *rational functional structure* and discussed the mechanisms by which this structure can be created. Therefore it was suggested that the development of the actual architecture of the product should emulate the same adaptability properties, hence calling the approach the *subordination of the physical structure to a rational functional structure*. Since a functional structure is in fact the nested segmentation of functions, the two ways of calling the proposed approach are conceptually identical.

The design etiquette which is encouraged in this approach has the following three principles. These characteristics are sought at the first few levels of decomposition of functions as discussed in Section 4.1.

- First, an adaptable product has a *modular* architecture, so that the required modifications of an adaptation task do not propagate throughout the product. The nesting of modularization generates a hierarchy of subsystems.
- Second, subsystems are *functional modules* which are designed to perform unambiguous and useful functions. The operation of a functional module is relatively independent from the product it serves, and its functional interaction with other assemblies in any product is the exchange of inputs and outputs.
- Third, subsystems are *autonomous* and self-contained so that they can perform their function independently from their working environment.

Section 4.2 discussed the challenges of applying this approach to the design of mechanical systems. This section describes the methods and guidelines which help in overcoming these difficulties and thus achieving a generally adaptable architecture in the design of mechanical systems.

Utilizing alternative technologies (software instead of hardware)

The function of many mechanical systems can be described by the output they generate from a given input [Pahl 1988]. The relation between the input and the output can be very complicated. For example, turning the steering wheel of a car causes the wheels to turn, at different angles for inner and outer wheels, and at different sensitivity ratios for different vehicle speeds. These functions can be achieved by mechanical means: a rack and pinion system turns the wheels, the geometry of the steering knuckle achieves the difference in angles of the two front wheels, and a governor adjusts sensitivity according to vehicle speed. These functions, which are the relation between inputs and outputs, could be performed in software if inputs were turned into computer data, and outputs were generated by electric motors (actuators) at computer command.

A technological enabler for the development of generally adaptable products is the digital and software control systems, which are becoming increasingly affordable. The phrase *by-wire* technology refers to various systems that have replaced conventional mechanical, hydraulic, pneumatic, and electro-mechanical systems with digital control systems. This technology creates a *soft* link between inputs and outputs of a system, where functions are performed via software and not by means of mechanical components. Inputs are processed by a computer that generates the appropriate signals to activate the electric-powered actuators which generate the appropriate output. The following figure shows the schematic diagram of the general design of such systems.

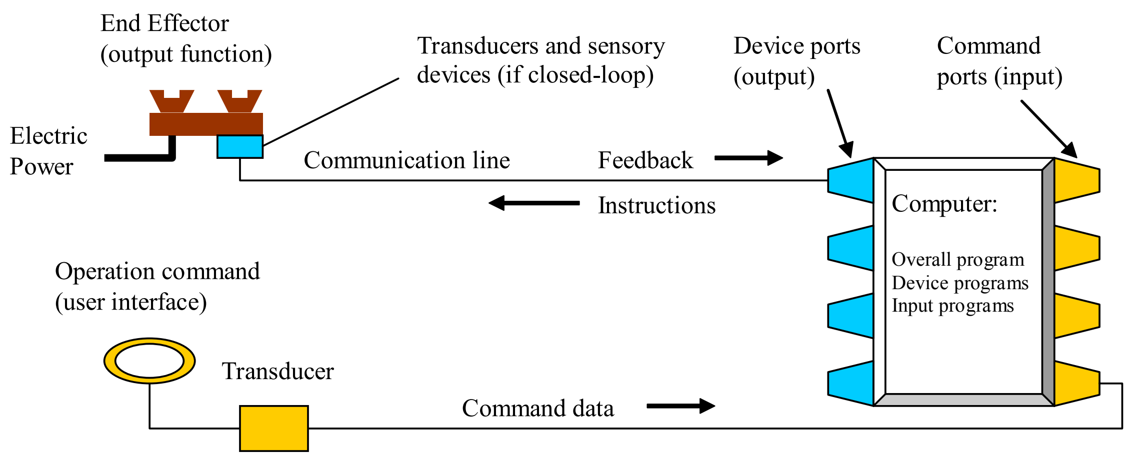


Figure 4. 9: The replacement of mechanical systems by "soft" electro-mechanical systems.

Autonomy of modules

Develop subsystems as autonomous modules. These modules do not depend on a particular configuration or on other components within a product for their proper functioning. They should contain their required elements within them.

Mounting small subsystems

From the above rule it can be deduced that a subsystem should be mounted on the system it belongs to "functionally". For example a car's radiator, which is mounted on the car's body due to spatial considerations, should be mounted on the engine so that the engine becomes autonomous in its function.

Functions meaningful and recurring

There are several ways of decomposing a given FR as discussed in Section 4.1. Among several scenarios, one should be chosen whose sub-FRs are more meaningful, widely usable, and recurring. The "meaning" of a function will be discussed in Chapter 6.

First develop input/output functions, and then develop internal mechanisms

This guideline can be deduced from the previous guideline. Since the purpose of a device is understood from its input/output functions, these functions are more likely to be meaningful. For instance, in the design of a vehicle an important function is accomplished by the 'seat', whose function is easily understood by the user. Internal components such as transmission or rack-and-pinion, though complex in their designs, are of no direct relevance to the user.

Smaller Sizes

Assemblies, while capable of delivering the required function, should be developed with the smallest size that can be reasonably achieved. This may require replacing mechanical systems with alternative technologies or reducing the size of mechanical parts when possible. Utilization of smaller sizes reduces the active spatial constraints and thus facilitates modifications and rearrangements during adaptation tasks.

Integral Flexibility

Use flexible elements whose flexibility is not related to modularity: hydraulic hoses, manufacturing adjustments, flexible shafts, wires, universal joints, etc.

Standardization

Increase the compatibility among subsystems through the standardization of systems and their interfaces. For instance in a subsystem which provides mechanical power, a shaft is a more standard and compatible form of output than a gear which requires its exact mating and adjustments. The interfaces among interacting mechanical components must be designed to provide for high levels of compatibility and flexibility. The proper design for adaptability and exchangeability of mechanical interfaces is discussed in the design of mechanical bus systems ([Gu 2002], [Gu 2003]).

Manufacturing Adjustments

Adaptability and flexibility can be increased by the use of *manufacturing adjustments*. These are tuning design features that provide for high tolerance in dimensions or positioning of parts, or they allow variations in other parameters such as voltage. This tolerance increases adaptability of the product, both in its design and its construction, to environmental variations (i.e. noise) and to varying service requirements. ([Otto 1994], [Lee 2003]).

Regular and Generic Forms

Use regular surfaces and generic forms which facilitate future alterations and amendments. Flat or cylindrical surfaces are preferred over sculpture surfaces; rectangular boxes are preferred over complex-shaped objects; and so on.

Physical Independence

Functional modules, which are functionally independent by definition, should also be made physically independent as much as possible. This guideline emphasizes the difference between general AD and the Independence Axiom proposed in [Suh 1990].

An Example: Adaptable Design of a Lock

A lock consists of two input/output functions. One is the intake of the user's signal indicating which state of the device is desired: locked or unlocked. The second is the actual locking action, which depends on the application of the lock. For example, in a lock that is used on a hinged door of a building, the locking action might be to drive a deadbolt between the door and its frame; and in a lock used in a revolving door, the locking action might be to prevent the main shaft from spinning.

If the lock is designed by the rules of general AD, its two functions should be performed by two independent and autonomous modules, and the connection between these modules must be made as flexible as possible. Figure 4.10 illustrates such a design, which utilizes the general electro mechanical system discussed earlier in this section. The user module receives the open/close command, which could be in the form of turning a key, swiping a card, punching a code, using remote control, etc., and transforms it into an electric signal through the appropriate transducer. This signal is sent to the action module, where through a relay it activates an electric motor for the locking action.

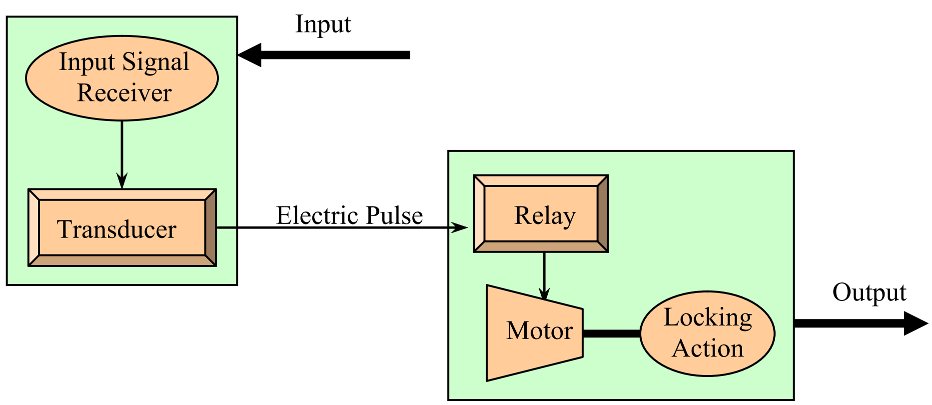


Figure 4. 10: The adaptable design of a lock.

Since the two functional modules in this design are connected by electric wires or wireless transmission, the lock system can be easily adapted to various configurations and installations. Therefore this design provides ‘product adaptability’. It also provides ‘design adaptability’ (variety) because modules are autonomous in their functions, thus various lock systems can be developed through the *morphological combination* of alternative designs for the two modules (Chapter 2). This is shown in Figure 4.11.

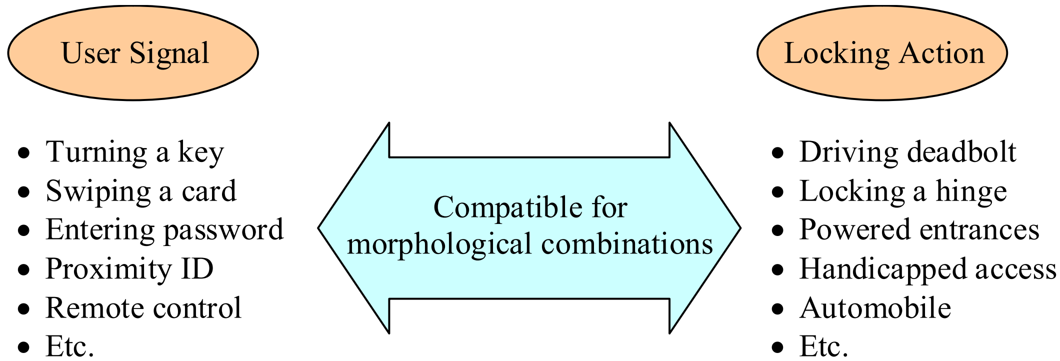


Figure 4. 11: Variety through the possibility of morphological combination.

The Frame and Function Architecture

In a service-based, as opposed to object-based economy, a product is a temporary assembly of functional elements in a configuration that is capable of delivering a temporary service [Ayres 1998]. There is no need for the product to exist as an object when its service is no longer required. Thus selling a product is in fact *selling use* ([Seliger 1997], [Lindahl 2001], [Seliger 2000]). This concept can be best observed in the new corporate model known as *virtual enterprise*. It is a company that is comprised of various functionally independent units that can provide their services regardless of the company they serve ([Goldman 1995], [Tolle 2003], [Gou 2003], [Bechler 1997]). A virtual enterprise is a temporary assembly of such autonomous units, often globally distributed and connected via communication networks, formed to exploit fast-changing opportunities such as making one particular type of product or delivering one particular type of service, then dissolving after the project is finished to allow the partners to find new partners.

Function-based segmentation of physical assemblies is the key method for the development of products that emulate the flexibility and adaptability of a virtual enterprise. A subsystem within a product is a relatively self-sufficient and autonomous module that corresponds to an obvious function in the functional hierarchy. These functions are generally understandable and describable, and more importantly, they recur in different designs. Such assemblies can be used in many overall products with different configurations and functions, as long as their interfaces have the appropriate inputs and outputs.

Therefore, in a service-based view towards engineering, a product is designed for a temporary use. It consists of a *frame,* as well as various functional assemblies. For any required overall function or service, a configuration for the product is designed, in which the frame determines the overall layout or embodiment of the product. Therefore a frame is designed to suit the embodiment requirements of the intended usage. A frame may be an abstract spatial order among physical assemblies, an actual physical structure such as a chassis or truss, or a combination of both. Then, various functional assemblies are added to the frame in order to achieve the desired functions. These modules might be mounted on a physical frame or on each other in the appropriate spatial order. Special components and product-specific assemblies, if required, can also be installed to make the product functional.

With this architecture, for any new application or operational mode the product can be adapted by the appropriate configuration of the frame and functions. An adaptation task involves the reconfiguration, replacement, and redesign of functional modules and the appropriate modification of the frame and special parts. The frame-and-function architecture possesses the properties of general adaptability discussed in this chapter.

4.5. Adaptable Design Methodology

If forecast information on future adaptations is available, it should be utilized. Thus in the overall AD methodology 'design for specific adaptabilities' has a higher priority than 'design for general adaptability'. This section describes the proposed methodology in this sequence.

Phase 1. Specific AD

- Define the original design problem (FRs).
- Identify the set of target adaptation tasks (Sp). This process utilizes forecast information on versatility, upgrading, customization, and variety.
- Develop a functional structure that includes both original FRs and the requirements of future adaptations (AFRs).
- Design the physical structure of the product according to the applicable methods and guidelines of specific AD. Depending on the methods used, subsystems might be developed as shared platforms, replaceable or interchangeable modules, optional add-on features, etc.

Phase 2. General AD

- In the absence of sufficient forecast information, assemblies should be designed as functional modules, beginning with modules that interact with the environment (input/output functions), then developing the internal mechanisms.
- The suggestions of specific AD overrule the suggestions of general AD.
- Functional modules should be developed so that they are as autonomous and

self-sufficient as possible.

Phase 3. Development

- Maintain the functional and physical independence between subsystems, especially between functional modules developed by the guidelines of general AD. Connect subsystems by means of soft interfaces such as wires and manufacturing adjustments.
- Whenever possible achieve functions by software, use alternative designs and technologies if necessary.
- Reduce sizes and utilize generic forms and regular surfaces in order to decrease spatial constraints. Also, utilize standard interfaces in order to increase interchangeability.
- Develop a spatial frame for the overall embodiment. Add the required subsystems to this frame.

Phase 4. Hierarchy

- Apply the above methodology to the development of individual assemblies. Depending on the complexity of a design, the segmentation process can be nested for a few levels, dividing the overall design into assemblies and dividing the larger assemblies into smaller assemblies. This results in a hierarchical structure of physical parts that conforms to the hierarchy of the rational functional structure.

4.6. AD and Other Life-Cycle Design Goals

The environmental benefit of AD stems from reducing production volume by having fewer products for the same amount of service. This can be translated to using fewer natural resources in preparing finished goods. In addition to adaptability, other environmental benefits might be sought when designing an adaptable product. These include: the use of non-toxic and environmentally friendly materials; design for recycling and reuse when applicable; design for manufacturing processes with low environmental impact on nonrenewable resources; and designing products that consume less energy and causes less pollution in their operation. Most of these environmental characteristics of a product are relatively independent from its overall architecture, which is determined by AD. They are determined by other design decisions such as the choices of materials and manufacturing processes, or the choice of solution principles for the operation of the artifact.

In addition to the reduction of environmental impacts, several other life cycle objectives may be sought in the design process. These include: quality issues such as performance, reliability, durability, and safety; design for manual or automated assembly; design for repair and maintenance; design for low cost manufacturing; and design for rapid product development. Most of these characteristics are also to a large extent determined by design specifications other than the product's overall architecture.

AD is applicable from the early stage of the design process until it determines the overall architecture and embodiment of the product at the conceptual level. As the design proceeds to more detailed stages, there is less freedom to make function-related changes in design and AD becomes inapplicable. On the other hand, other life cycle design issues mentioned above are less relevant at the beginning of the design process because little is known about parts, material types, repair frequency, etc. As design proceeds to further levels of detail, more detailed information is available and these life cycle issues become more relevant.

Therefore, adaptable design ought to be performed at earlier stages of design than those life cycle design issues which depend on detailed design specifications. This statement suggests a sequence in procedures and eliminates the conflict, which is fortunate. That is, adaptable design must be performed first, and then when further detailed information is available, other life cycle aspects can be considered. Iterations, as an inherent part of any design process, might be inevitable.

Figure 4.12 illustrates this sequence. The dotted curves indicate that the driving objective of modularization at the beginning of the design process is adaptability, which is based on the segmentation of functions. Towards the end of the process other life cycle goals related to detail specifications become more applicable.

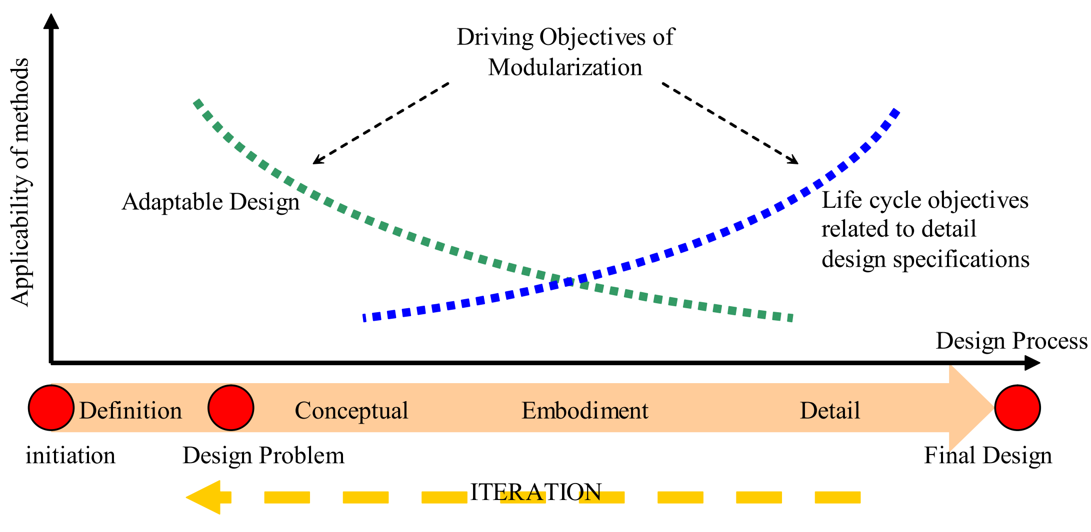


Figure 4. 12: The applicability of adaptable design and other life cycle design in the design process.

4.7. Summary

This chapter described the *design hierarchy* and the *decomposition* mechanism by which this hierarchy is generated. In an ideal scenario, this process results in a *rational functional structure* with a *causal* relation among its elements. This relation makes such a structure suitable for adaptability. Section 4.1 concluded that the imitation of this structure in the actual architecture of a product is a logical approach towards achieving general adaptability in designs. Section 4.2 discussed the inherent properties of mechanical design that hinder this imitation. These properties stem from the *structural connectivity* among solid parts, and more importantly, from the *functional ambiguity* of subsystems which are created through the conventional mechanical design process. Section 4.3 introduced a measure for adaptability based on the amount of 'saving' which is achieved through adaptation. These savings were measured by the *information content*, which is asserted to represent 'total costs' more accurately than monetary value does. Section 4.4 discussed the methods and guidelines for both *specific AD* and *general AD*. Section 4.5 presented the overall methodology of *design for adaptability* in which specific AD was given a higher priority than general AD. Section 4.6 showed how this methodology can be extended to accommodate *life cycle design*.

The next chapter will present the conceptual design of a few mechanical systems which are designed according to the overall methodology and the guidelines of this chapter.

Chapter 5: Examples

This chapter discusses the adaptable design of a few mechanical systems. These examples are conceptual designs; and for simplicity they are not designed for life cycle objectives. Section 5.1 discusses an example of *specific AD* in which a product is designed for predetermined adaptations. Section 5.2 presents two mechanical systems which are designed for *general adaptability* without targeting particular adaptations. Section 5.3 first describes an example of design for specific adaptability from the design literature, and then designs the same system for general adaptability in order to demonstrate the differences between *specific AD* and *general AD*.

5.1. Specific AD: Versatile Bicycle Accessories

Among the four categories of specific AD discussed in Chapter 3, *design for versatility* is the most specific category. Design for versatility has the most forecast information, and specifically designs a product for a predetermined set of adaptations (Chapter 3, Figure 3.8). A versatile design has been chosen as the example of specific AD in this section. This example is the versatile design of bicycle accessories similar to the Master Lock design shown in Figure 3.10. This is a case of *product (user) adaptability*; and adaptations from one function to another occur frequently and reversibly. Therefore the final design should allow these adaptations to be performed easily by the user.

5.1.1. The Design Process

The design process follows the AD methodology discussed in Chapter 4.

Step 1: Original FRs and Additional FRs

Various accessories are available for bicycles: locks, carrier racks, fenders, bottle holders, tools, storage compartments, etc. These accessories can be attached to a bicycle in several ways. There is a level of redundancy if these accessories are individually installed, hence suggesting the possibility of a versatile design which can perform the function of multiple accessories.

After comparing various *functions* of bike accessories, the *carrier rack* and the *splashguard* have been chosen for adaptable design. The reason for this choice is the similarity between the functional structures of these two devices. Their initial functional requirements are:

Carrier rack:

- FR1: Attach firmly to the bicycle.
- FR2: Hold a load of up to 10kg.
- FR3: Do not interfere with the normal operation of the bike.

Splashguard:

- FR1: Attach to the bike.
- FR2: Protect the rider from water thrown by the rear wheel.
- FR3: Do not interfere with the normal operation of the bike.

Figure 5.1 shows the functional structures as well as solutions for both the carrier rack and the splashguard. *Physical functions*, which require physical components as their solutions, are denoted by F; and non-physical functions are denoted by NF. The comparison of these structures reveals a common physical function, which is the attachment of an accessory to the same location on the bicycle. Therefore, one module could perform this common function in both applications.

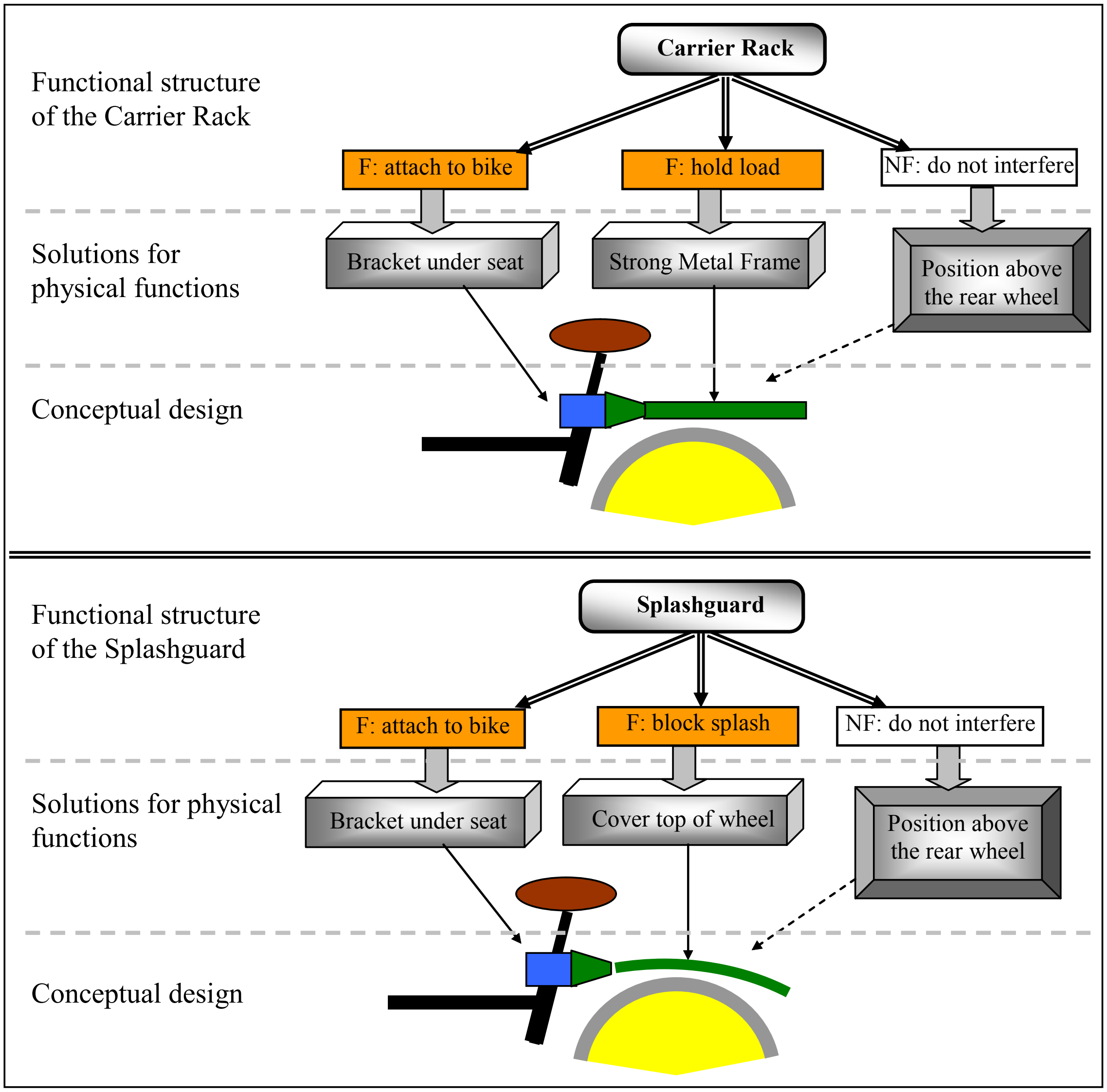


Figure 5. 1: The functional structures of a carrier rack and a splashguard.

Also, the comparison of the *physical embodiments* of bicycle accessories reveals that there is a similarity between the metal frame of the carrier rack and a U-lock. This suggests the possibility of including the U-lock in this versatile design. Therefore, its FRs must be added to the FRs of the previous two devices in the list of AFRs (Additional FRs) for adaptable design. The FRs of the U-lock are as follows:

- FR1: Attach to the bicycle when not in use.
- FR2: Lock the bicycle.
- FR3: Provide security against theft.
- FR4: Do not interfere with the normal operation of the bike.

The schematic diagram of the functional structure and a conceptual solution for this product are shown in Figure 5.2.

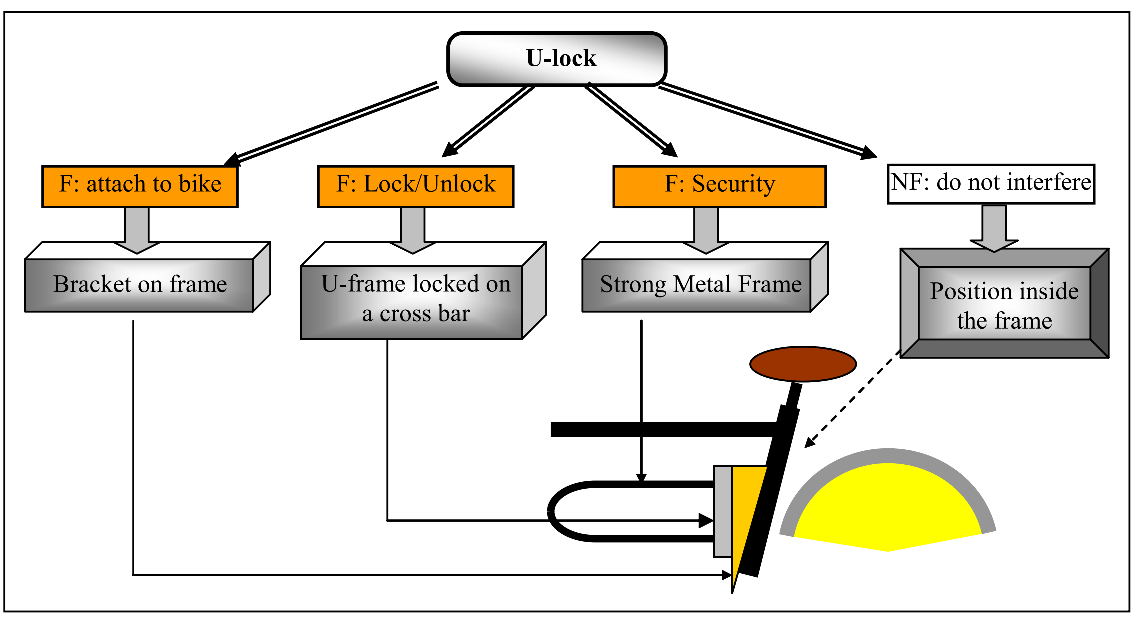


Figure 5.2: The functional structure of a U-Lock.

Step 2: Three Individual Designs

This step develops individual solutions or conceptual designs for individual applications. In this example these are known from existing designs. The conceptual designs for the splashguard, the carrier rack, and the u-lock are shown in Figures 5.1 and 5.2. These designs represent the Ideal Initial States for individual designs, as explained in Section 2 of Chapter 4. The ideal initial states are needed in order to assess the adaptability of final design using Equation (8) of Chapter 4.

Step 3: The Conceptual Design of the Adaptable Product

Figure 5.3 shows an adaptable design for this example. The assembly of the overall product is shown on top, and the solid models for major components are shown on the bottom. The U-shaped frame is designed for quick attachment and is made strong to provide both lock security and rack strength. The deadbolt is designed asymmetrically, so that it slides into the bracket only when its orientation for the insertion of the U-frame into holes is correct. The deadbolt also includes a square hole for the attachment of wrench bits, and a patterned hole at the end for the attachment of screwdriver bits. These extra features can be created at negligible cost. The bracket is designed symmetrically so that the deadbolt can be inserted from either side. Its design includes tapering ribs that transfer the load of the rack to the seat bar, thus achieving strength without adding to weight. The top of the bracket is designed as a simple flat surface in accordance with the AD guidelines. This flat surface is likely to facilitate the addition of extra features to the design in the future. By attaching a fender to the same bracket, this design can perform all three sets of FRs.

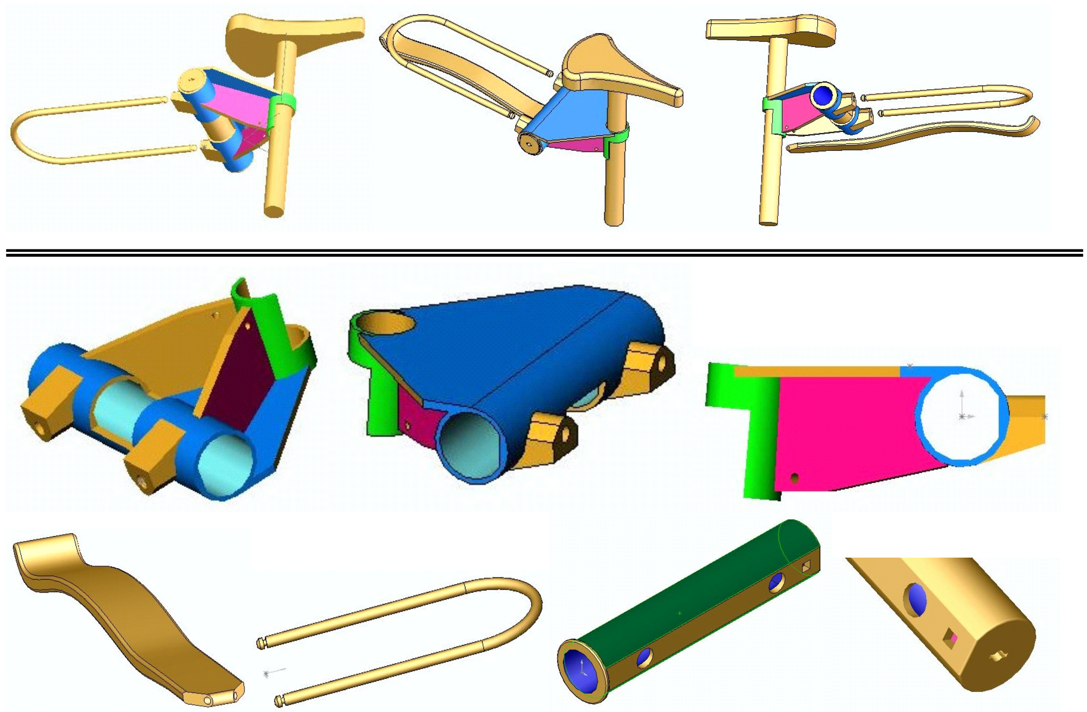


Figure 5.3: The design of an adaptable bicycle rack.

Step 4: Adaptability

In the measurement of adaptability for this example three simplifying assumptions are made. First, it is assumed that the probability of occurrence for all adaptation tasks is 1, because the product will be used as a rack, a lock, and a splashguard during any given service period. Second, it is assumed that adapting the configuration of the product from one usage to another, for example from lock to rack, does not require noticeable effort. Therefore the information content of performing any adaptation task is zero. The information contents for various tasks involved in Equation (8) are representative of their complexities and costs. The above adaptable design is more complex and costly to materialize than any of the three individual products. However, the third assumption is that these higher initial costs are negligible. The effects of such higher costs will be discussed in another example in this chapter (Section 5.3.1). With these three assumptions, Equation (8) of Chapter 4 can be used to measure the adaptability of this product as follows:

Adaptability= (1-0) + (1-0) + (1-0) = 3

This means that the adaptable product designed above will replace three products which would be needed for the same service.

5.1.2. Discussion

The final design in this example is similar to an existing product. This is not unexpected as *specific AD* represents the existing design methods. This example demonstrates the fundamental element of specific AD which is to build predetermined adaptabilities into the product during the design process. It can be seen that performing predetermined adaptations in the final design is very easy. This design, however, is difficult to adapt to unforeseen changes. For instance, this design cannot be adapted to a new type of lock such as a chain lock.

5.1.3. Other Examples of Specific AD

Several other examples of specific adaptability were presented in Chapter 3. The Sony video cameras and Ford automobiles are examples of product *variety* (Figures 3.4 and 3.8). Both of these designs, unlike the bicycle accessory designed in this section, are instances of *design (producer) adaptability*, in which the adaptation of the same basic design for the production of different models is performed by the producer.

An example of specific AD for *upgrading* is the design of personal computer systems. Various parts such as the memories and processors undergo rapid technological obsolescence. These components are designed as easily-detachable modules. Therefore, the product can be upgraded by the user who can have these parts replaced easily, hence extending the product's life. Upgradeable computers may also be considered an instance of specific adaptability for the producers, who can efficiently upgrade their products and constantly utilize the state-of-the-art technology in their models. Some other instances of specific AD such as the design of modular robots were discussed in Chapter 2.

5.2. Examples of Design for General Adaptability

General AD aims to subordinate the physical structure of a product to its *rational functional structure*. The subsystems of such a product are autonomous functional modules; that is, each module independently performs a meaningful function corresponding to the functional structure. Since such ideal architecture may not be practically feasible, the guidelines of AD help the designer develop products which are closer to this architecture. These guidelines suggest that the development of modules begin from the input/output functions, which determine the purpose of a device, and that other functions be achieved by software instead of mechanical components where possible. The guidelines also encourage the use of flexible connections, standard interfaces, and generic forms as discussed in Chapter 4.

Thus, the main process of *design for general adaptability* is to develop a rational functional structure and then develop independent modules for each FR in this structure with the help of the guidelines. There are many ways to develop a functional structure for a design problem, and the choice reflects the perception on the designer and determines the final outcome. It can be seen that this process does not depend on forecast information regarding future adaptations. Therefore, unlike the previous section, the examples in this section do not target predetermined adaptabilities.

This section presents two examples of general AD. They are conceptual designs for the purpose of demonstrating the basic procedure of design for general adaptability. The details of these systems are not discussed and they are not designed for financial or aesthetics criteria. When applicable, the technological feasibility of a design concept is shown using similar industrial applications.

5.2.1. The Adaptable Design of a Hydraulic Jack

In solution-neutral terms, this example is the design of a mechanical device that amplifies the user's muscular force (input) to the required level (output). In general AD, these input and output functions should be designed as separate modules, and the connections between them must be made as flexible as possible.

Figure 5.4 shows a few conceptual designs for manual force amplification. The first four designs utilize solid mechanical connections such as gears or chain-and-sprocket. These designs impose fixed spatial relations between the input and the output. In the hydraulic systems shown on the bottom, however, the input and output functions are performed by independent modules (pump and jack). The connection between these two modules is a hydraulic hose, which is more flexible than solid parts. Thus, these hydraulic systems are more adaptable than the other designs in the figure.

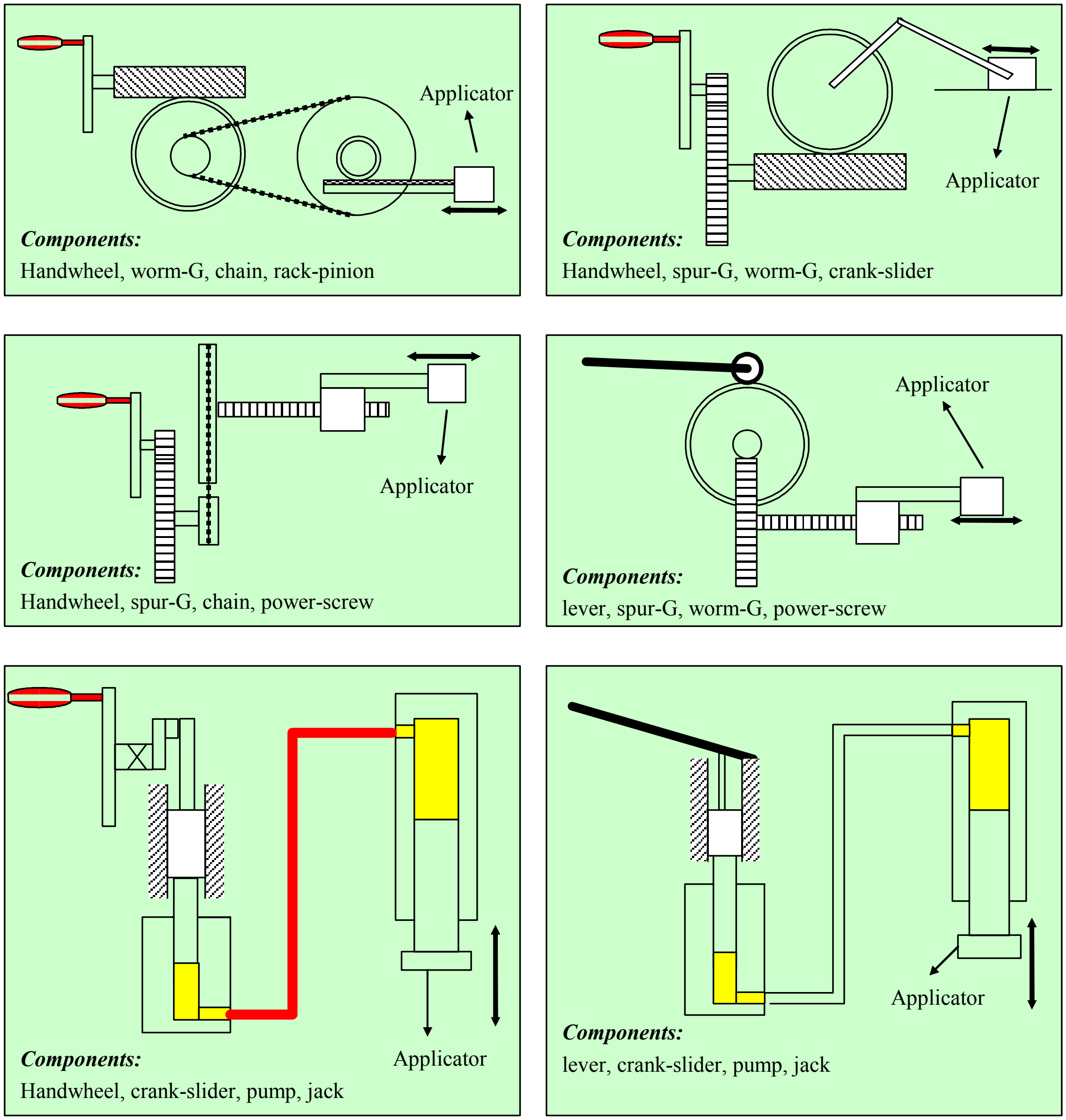


Figure 5.4: Conceptual designs for a manual force amplifying device.

Figure 5.5 illustrates a versatile design for a manual hydraulic pump. In order to be able to utilize this pump in various applications, it is designed to operate at two different speeds for different loads. The top of the figure shows the main pump body with its hydraulic conduits, and the double-diameter plunger assembly. The bottom of the figure shows the assembly of the double speed manual pump and its operation mechanism.

When the ‘selector lever’ shuts the bypass, the pump works at high speed and low pressure. The lifting of the plunger sucks the fluid into both cylinders, and pressing it down delivers the liquid from both cylinders through the output hose. When the ‘selector lever’ opens the bypass, the bigger cylinder becomes idle because it is connected to the fluid tank during both upward and downward strokes. In this case, only the small plunger is active, resulting in higher force but lower speed.

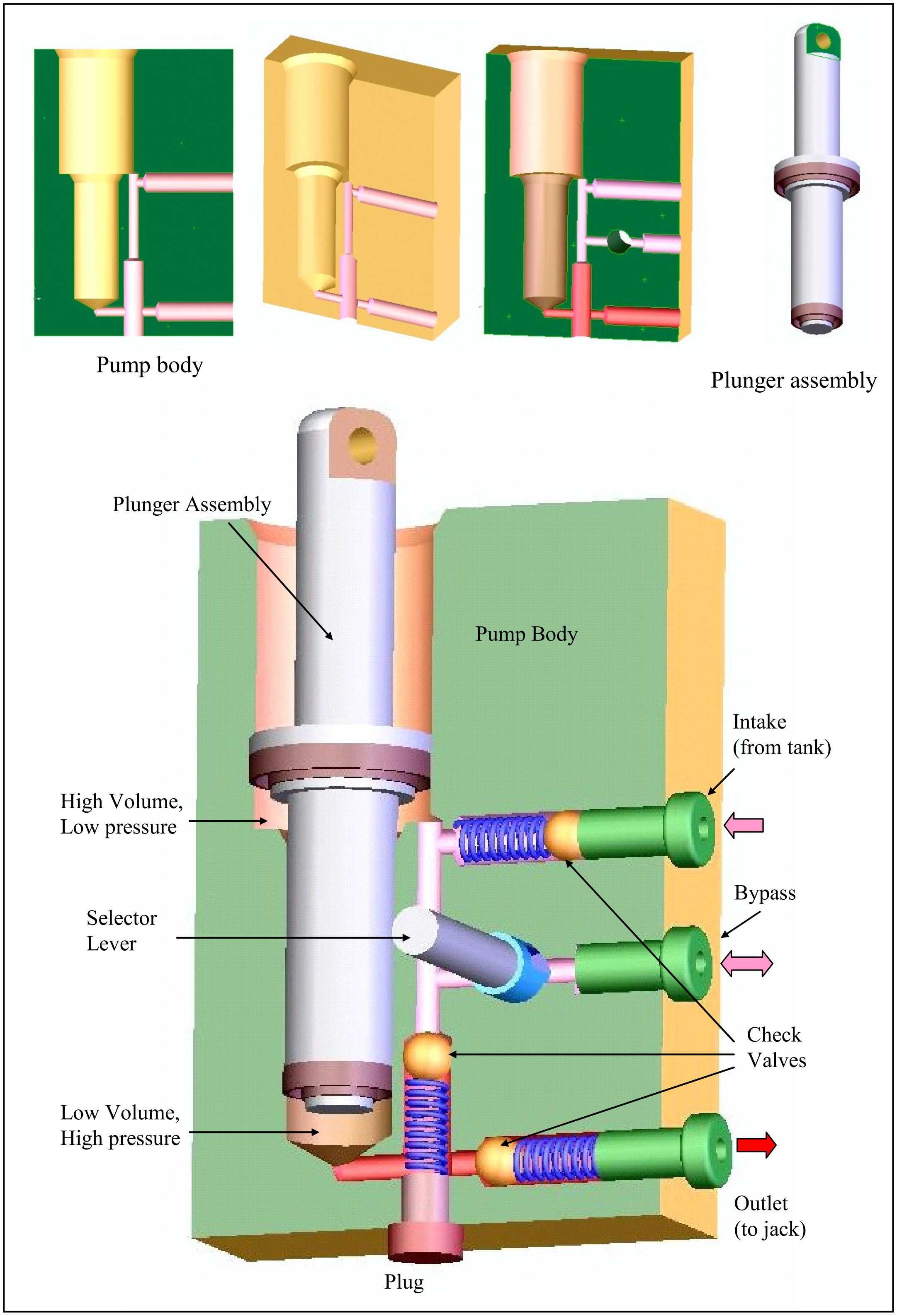
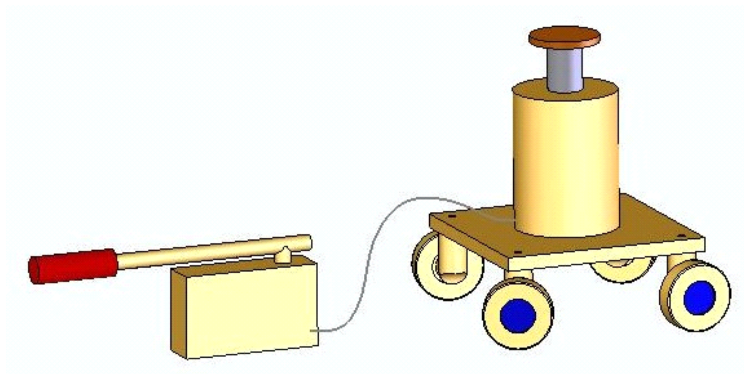


Figure 5.5: The design of a double-speed manual hydraulic pump.

Figure 5.6 illustrates the adaptable design of the hydraulic jack assembly. The pump is mounted inside a fluid tank. This tank is a rectangular box and supports the pump handle on top. A hose connects the pump and tank assembly to a hydraulic jack, which is mounted on a small trolley so that it can be moved to the desired location under a vehicle. The bottom of this figure shows two conventional hydraulic jacks.



Adaptable Hydraulic Jack

Hydraulic Bottle Jack

Hydraulic Garage Jack

Figure 5. 6: An adaptable design and two conventional designs for hydraulic jacks.

Discussion

In this example the general adaptability of a system is increased through the development of independent functional modules, and without specifically preparing the product for predetermined adaptations. In the proposed design, the hydraulic pump is an independent module which can perform its function (delivery of hydraulic energy) in many circumstances. For example, it can be used to operate an automobile jack, a press, or a metal cutting tool such as the *Jaws of Life* used in emergency rescue operations. Similarly, the jack is designed as an independent module that can take its hydraulic input from the above pump or other sources of hydraulic pressure such as an electric pump. Therefore, this system and its modules can be adapted to various circumstances. In contrast, the conventional designs shown in the Figure 5.6 are usable only in their normal operational mode.

5.2.2. The Adaptable Design of a Vehicle

The function of a modern vehicle goes well beyond the simple provision of mobility. In the design of vehicles many complex performance requirements are considered such as ergonomics, steering and road handling, suspension and dynamic stability, passenger comfort, collision safety, fuel efficiency, pollution control, aesthetics, and so on. The example in this section does not consider these requirements, and only designs a vehicle for its primary function of *transportation*. The conceptual designs discussed in this section are developed by the guidelines of design for general adaptability, and are intended to demonstrate the application of the frame-and-function architecture.

In this section, first the functional structure of a vehicle is developed; then for each function in this structure an independent physical module is designed. For an operational mode a vehicle is assembled from these modules in a configuration (frame) that suits a temporary service. In this section several vehicles for different operational modes are conceptually designed through the assembly of functional modules.

The Functional Structure

As suggested in the Phase 2 of the methodology in Chapter 4, the development of the functional structure begins by identifying the intended input/output functions. These functions represent the purpose of a vehicle and they are thus the main design objectives. In this example the following functions are considered for a vehicle:

- Mobility on land (Traction)
- Positioning passengers
- Driving and control
- Source of power

There are many other input/output functions that are not considered such as protecting occupants from the outside environment, carrying loads, attaching optional devices, and so on. A vehicle's design should also satisfy many requirements which are not *physical functions*, such as safety, cost, aesthetics, and simplicity. These requirements are not discussed in this example. Figure 5.7 shows the simplified functional structure of a vehicle based on the above four functions.

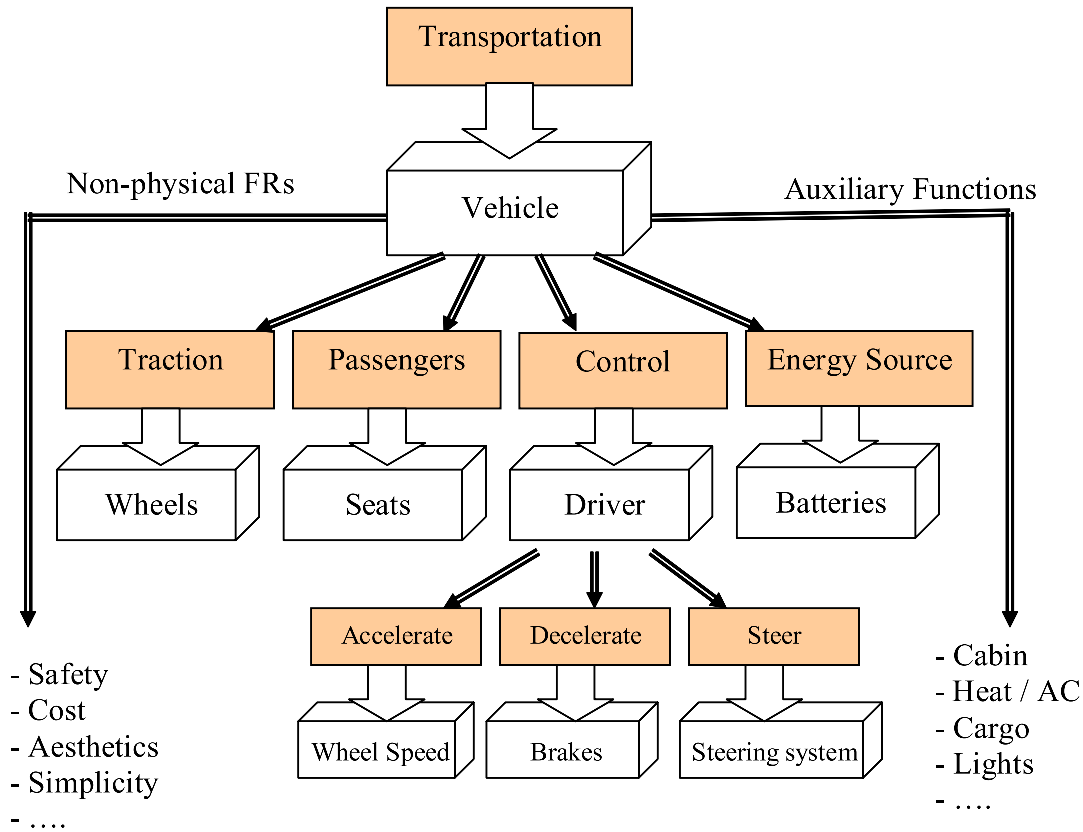


Figure 5. 7: A functional structure for a vehicle.

Functional Modules

The next step is the development of independent modules for performing each of the four functions in the functional structure.

Function One, Traction; Solution, Motorized Wheels

For the function of land traction the chosen solution in this example is a motorized wheel. This module has two sub-functions, the first is to provide rotational mechanical power to the wheel and the second is to provide traction on terrain. Thus it consists of two subsystems: an electric motor and a rim/tire assembly.

The electric motor is designed to have its stator and electronic control systems mounted on a stationary inner shaft, while its outside housing is the rotor. The advantages of having the rotor on the outside are: the possibility of mounting the motor inside the wheel, access to motor from both sides, and using the body of the motor as the hub of the wheel thus saving in material. Figure 5.8 shows the operation of the motor within the vehicle's overall system. The figure also shows the system boundary for the motor module. As the figure shows, the motor technology with dynamic reconfiguration for performance optimization is commercially available. The function of this module is to receive electric power and operation command signals, and generate mechanical power with characteristics that are demanded by the command signal. Therefore the motor has two input ports, which can be designed as sockets, and one output, which is the rotary outside hub.

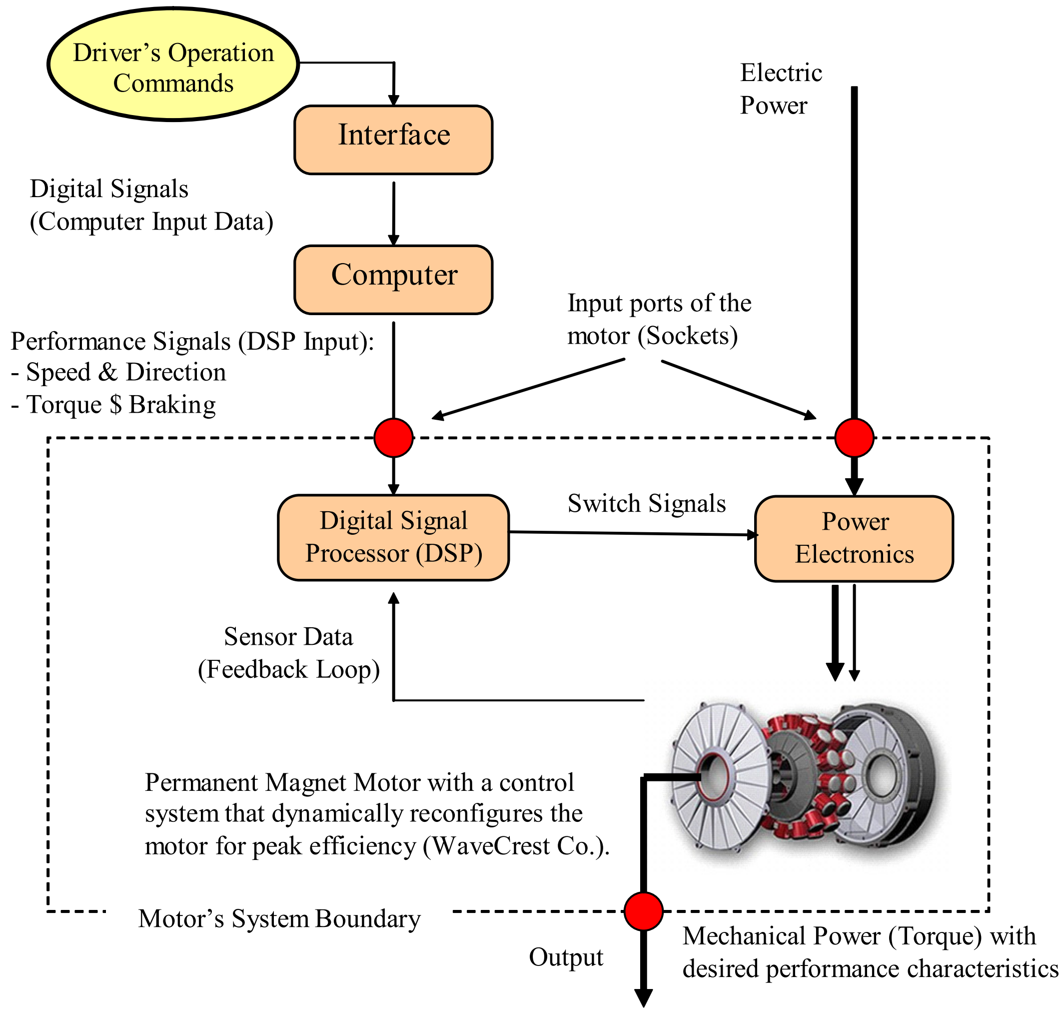


Figure 5. 8: The operation of a wheel motor (picture courtesy of WaveCrest Co.).

Figure 5.9 shows an embodiment design for this motor. This design has a splined outside hub for mounting different rims and tires. It also has mechanical, electrical, and electronic connection ports for the coupling of two or more motors. The figure also shows three types of rim/tire assemblies. Any of these can be mounted on the motor for different applications as shown in the bottom of the figure. The wheel on the right is wider and includes two electric motors in its hub. An assembled wheel is an independent unit for the traction function, and it is connected to the rest of vehicle through two sockets, one for operation signals and the other for power.

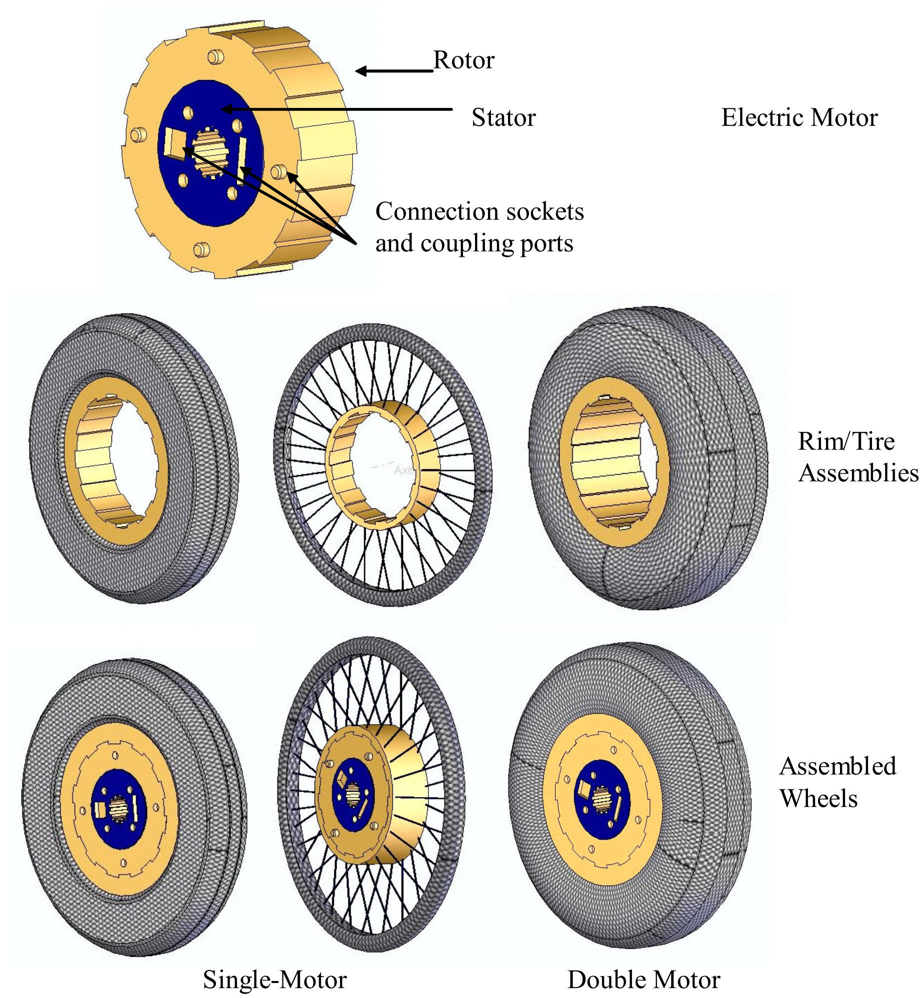


Figure 5. 9: Electric wheels designed as independent functional modules.

Function Two, Positioning Passengers; Solution, Seats

The seat should be design as an independent module that can be used for its function in any configuration. There are many such designs available, such as the two seats shown in Figure 5.10.

Figure 5. 10: Seats for the function of 'positioning passengers'.

Function Three, Control; Solution, Electronic Driver Interface

The three functions of controlling a vehicle are: increasing speed, reducing speed (braking), and steering. The acceleration function in a motorized wheel is readily available because the torque and speed of the motor can be directly controlled by the driver. The braking function, including *regenerative braking*, can be implemented through the *"by-wire"* braking technology. In automotive industry the term "by-wire" symbolizes the tendency towards utilizing electro-mechanically driven devices instead of mechanical, hydraulic or pneumatic components [Doriben 2003]. The *brake-by-wire* design is available in the literature and is not discussed here. Therefore, this section only discusses the steering function and the design of an innovative steering system.

The functional structure of a steering mechanism consists of two input/output functions: the driver's steering interface (input), and the actual turning of the wheels (output). This section first briefly describes the conventional design of steering systems as well a more adaptable design recently introduced in the industry. Then it discusses the conceptual design of a steering system which is designed for general adaptability.

The Conventional Steering System

In a conventional steering system the driver's interface is a steering wheel, which is connected to a rack-and pinion via a shaft (Figure 5.11). The linear motion of the rack is transformed into the pivotal motion of wheels by the steering knuckles. Thus in this system the input and output functions are connected via mechanical links, which impose various spatial and structural constraints in the design of a vehicle. For instance the location of the steering wheel is coupled with interior design, and the designs of the rack-and-pinion assembly and steering knuckles are linked with the design of chassis. Therefore, modifying and adapting this system is difficult. A conventional steering system is typically used in its nominal operation mode only.

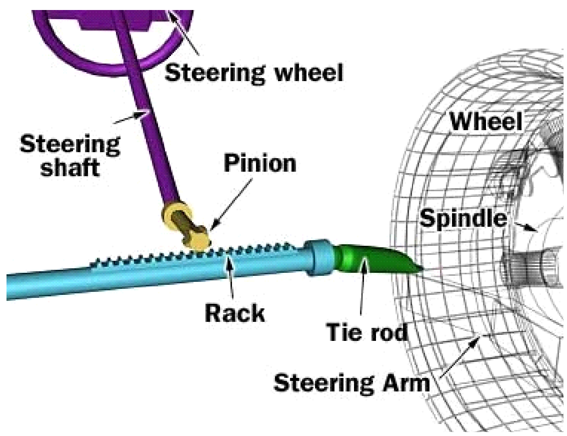


Figure 5. 11: The conventional steering system.

By-Wire Steering

The separation of the driver input function from the rest of the steering system can be seen in *by-wire steering* (Figure 5.12). In this design the input function is accomplished by an independent module which is connected to the rest of the system by wires. This module consists of a driver interface (wheel, handgrips, etc.), position sensors, and the electronic/software systems that generate the appropriate signals for the pinion's servo motor. The rest of the steering system includes various feedback sensors for closed-loop control; a servo motor for driving the pinion, and mechanical components such as the rack and steering knuckles similar to the mechanisms of a conventional design.

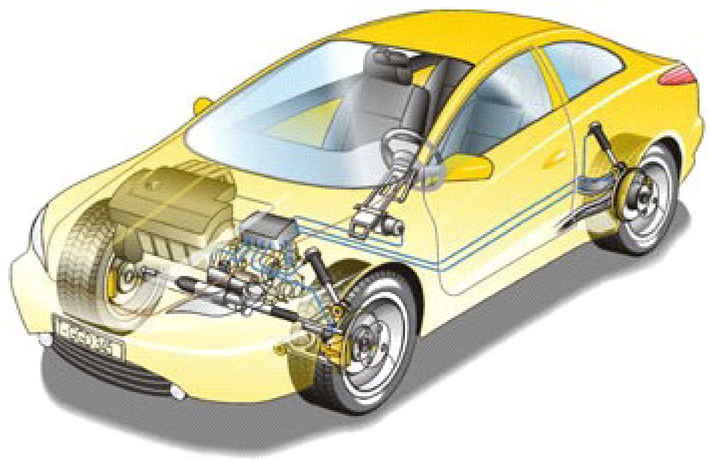


Figure 5. 12: By-wire steering. (Picture courtesy of SKF)

Figure 5.13 shows the driver interface module designed by GM for the by-wire steering of their 2002 concept car called AUTONOMY. It can be seen that the steering handle can be mounted in different locations. Therefore, the position of the driver is not constrained, resulting in flexibility and adaptability in the design of an automobile.

Figure 5. 13: The driver steering module in AUTONOMY (Courtesy of GM).

A Generally-Adaptable Design: Servo Steering

The by-wire steering system is more adaptable than the conventional design because its driver interface is designed as an independent module. The rest of the system, however, is as non adaptable as the conventional design. Figure 5.14 shows how the adaptability of the design can be further increased. Diagram (a) represents the conventional design. Diagram (b) shows the by-wire steering system, in which the steering shaft and the booster are replaced by a servo motor that turns the pinion, while the rack-and pinion and steering knuckles remain intact. Diagram (c) shows our proposed design in which the wheels are independently turned by two separate servo motors. This design, called *servo steering* in this thesis, eliminates the need for the rack-and-pinion, hence removing the structural dependency between the right and left wheels.

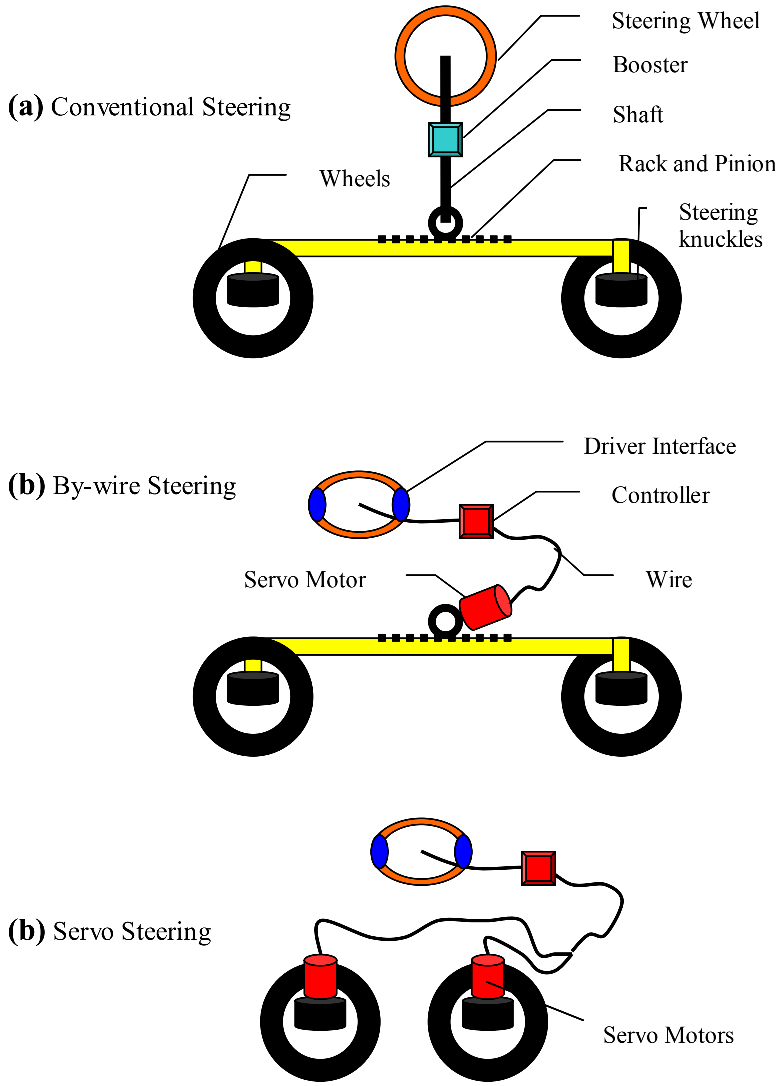
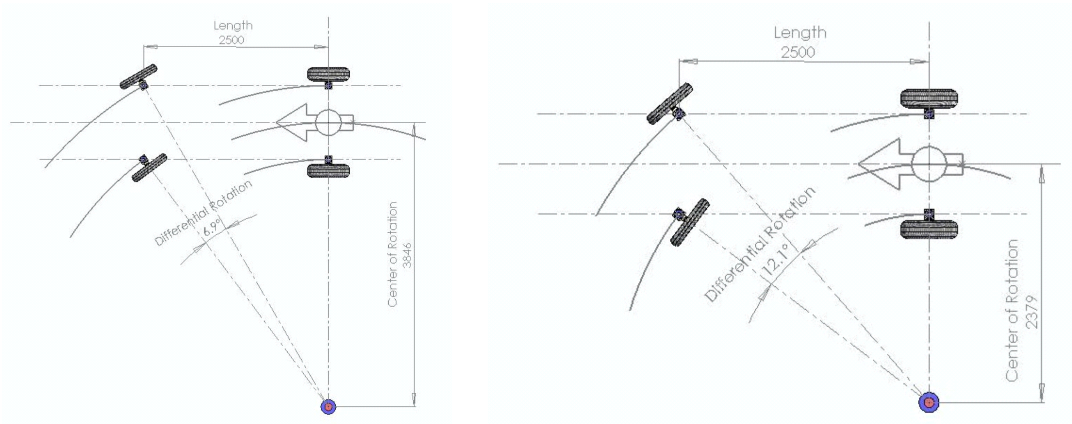


Figure 5. 14: Increasing the general adaptability of steering systems.

The mechanical separation of two steered (front) wheels requires the synchronization of the turning angles for the two wheels, which can be easily achieved by software. This is illustrated in Figure 5.15. When the driver steers, the right and left front wheels turn at different angles so that their axes intersect on a point located on the rear axis. This point is the "*instantaneous center of rotation*" for the vehicle. The real-time calculation and implementation of these angles is an easy task. Such technology has been utilized in CNC machines for decades. The servo steering system can also be applied to any number of wheels, which creates new possibilities for steering a vehicle (Appendix 2).



Front steering at a small angle | Front steering at a large angle

Figure 5. 15: Calculating rotation angles for the right and left wheels.

Therefore, in the servo steering design both the driver's interface (input) and the turning mechanism for each individual wheel (output) are developed as independent functional modules. These modules perform a meaningful function according to the functional structure of a vehicle, and the connections among them are flexible wires. Further, all mechanical links between the input and output functions are replaced by software. It can be easily seen that this design imposes few constraints on the vehicle and thus facilitates modifications and adaptations.

Function Four, Energy; Solution, Batteries

There are several solutions for an onboard source of power: internal combustion engine, gas turbine, fuel cell, rechargeable batteries, solar energy, and so on. All these solutions are valid and currently used in various vehicles. For example diesel generators or gas turbine generators are used in train locomotives, solar panels are used in recreational light vehicles, and batteries or fuel cells are used in electric cars. In this section the battery is chosen as the solution. Figure 5.16 shows the concept of a modular battery for this example. The batteries have regular shape and flat surfaces and can be stacked up in various geometries to obtain the required voltage or power.

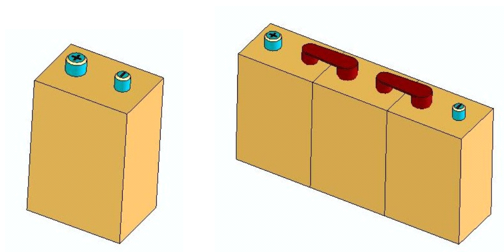


Figure 5. 16: Modular battery cells.

Frame (Chassis)

The chassis in this example is a spatial frame for the assembly of modules in an appropriate configuration. This section discusses the adaptable design of a chassis for a slow-moving vehicle such as a lunar vehicle or an AGV (Automated Guided Vehicle). In this design, adaptability is achieved through the structural modularization of the chassis, which can be designed as a *space frame*. A space frame is a three dimensional truss consisting of *links* and *joints.* Figure 5.17 shows the space frame elements designed for the chassis. The links are metal tubes of several standard lengths with a bolt assembly on each end. The tube lengths are designed to allow the construction of various geometries through primitive triangular patterns. The tubes in this design also have holes for passing wires and installing sockets. The bolt assemblies are designed to allow a bolt to turn without having to turn the tubes. The joints are forged meal balls that have several threaded holes at frequently-needed angles for the attachment of bolts. A simple frame in the bottom of the figure illustrates how the links are connected through joints to obtain triangular patterns.

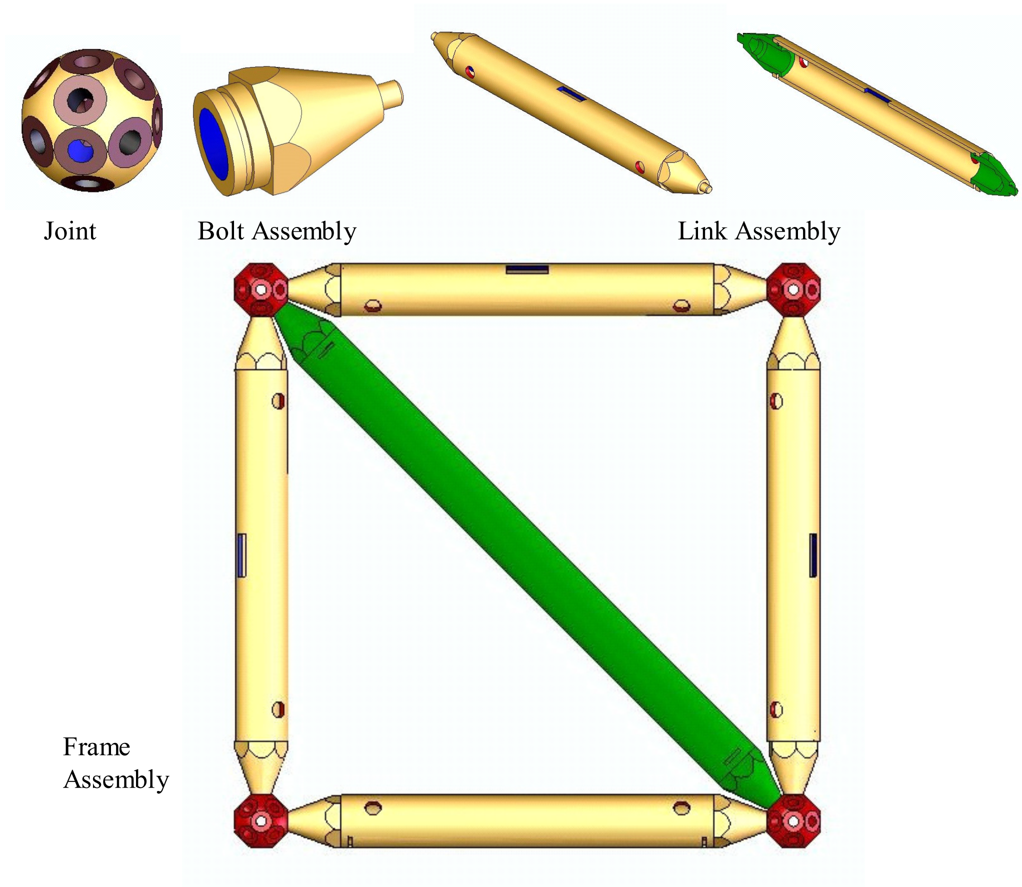


Figure 5. 17: The space frame elements designed for this example.

In this design, links and joints are independent functional modules that perform their functions regardless of the configuration they belong to. With an inventory of a few types of joints and a few sizes of links, a large number of frames with different sizes and configurations can be developed. A space frame structure is thus an adaptable design; it can be easily dismantled and reconstructed for new requirements. A space frame chassis can be easily modified in width, length, height, location of wheels, and the load-bearing configuration. Figure 5.18 shows a sample chassis designed by space frame elements. This design is a two-dimensional frame for better illustration; an actual chassis is a three dimensional frame so that it can carry vertical loads. The corner brackets are for the installation of wheels. They are designed as moment-bearing solid pieces so that they can convert the twists and forces of wheels into linear forces at the brackets' contact joints with the space frame.

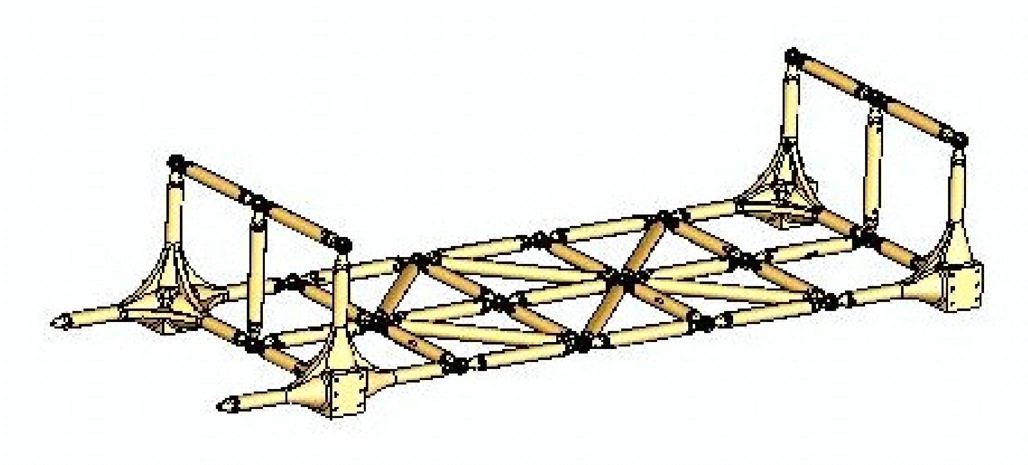
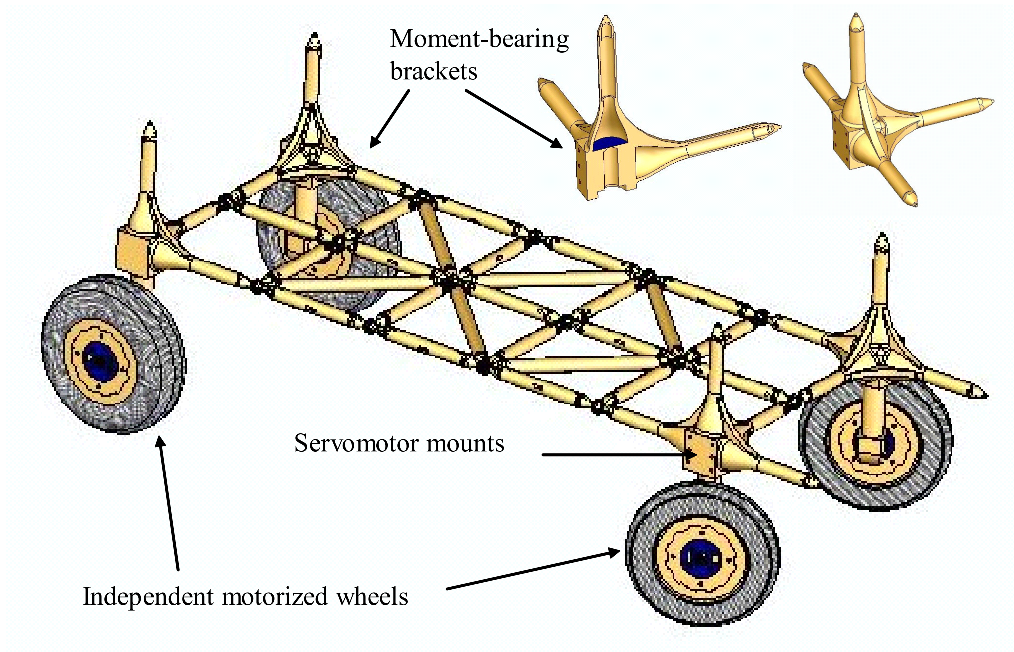


Figure 5. 18: A space frame chassis.

Vehicles

With the assembly of functional modules on various frames, various vehicles for various applications can be created. Figure 5.19 shows two sample configurations for cars. The batteries, seat, driver control and other modules can be mounted anywhere on these chasses and connected to the rest of the system by wires. The tubes in the space frame are equipped with holes for passing wires and sockets for plugging various control ports. The servo motors for the servo steering system can be mounted on any bracket; thus any wheel can be steered if required.



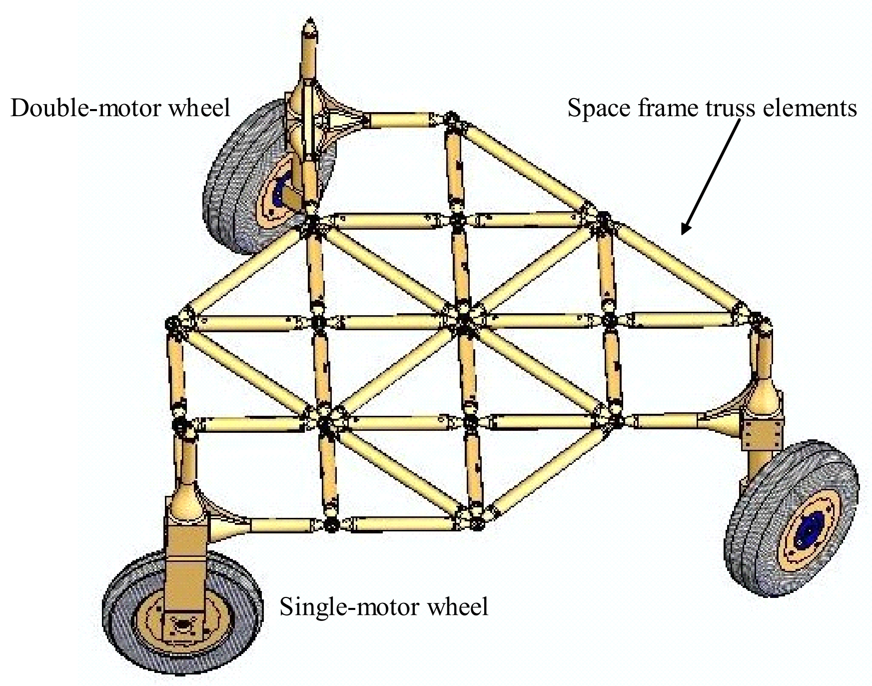


Figure 5. 19: Adaptable car configurations.

Figure 5.20 shows a few other examples of vehicles which can be constructed using the functional modules discussed above. The vehicles in this figure have application-specific frames which are not constructed from the space frame components. Many other vehicles with different configurations are also possible, such as golf carts, tandem bicycles, trucks, etc.

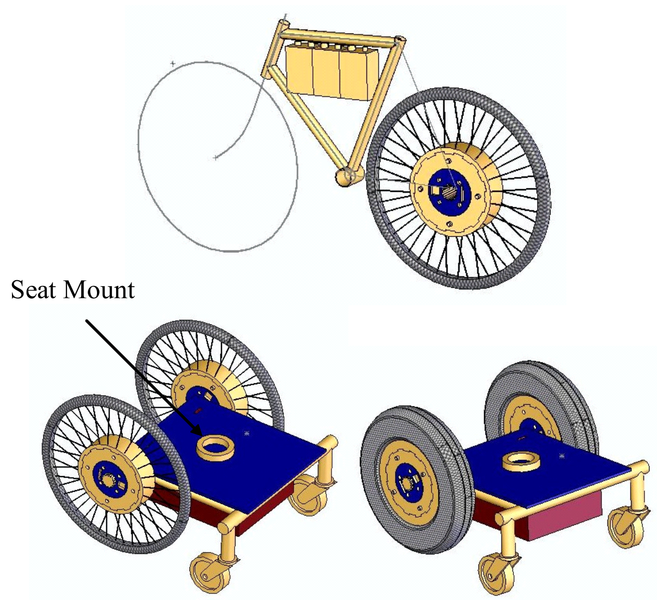


Figure 5. 20: Other types of vehicles that utilize the functional modules.

Discussion

Although the designs of this section are simple and only consider the basic functions of a vehicle, they demonstrate the ideas and strategies of design for adaptability. Figure 5.21 shows the rational functional structure of a vehicle in which functions are represented by their chosen solutions not in solution-neutral terms. The figure also shows a schematic diagram of the vehicle design. This figure illustrates the one-to-one correspondence between the functional and physical structures in this design.

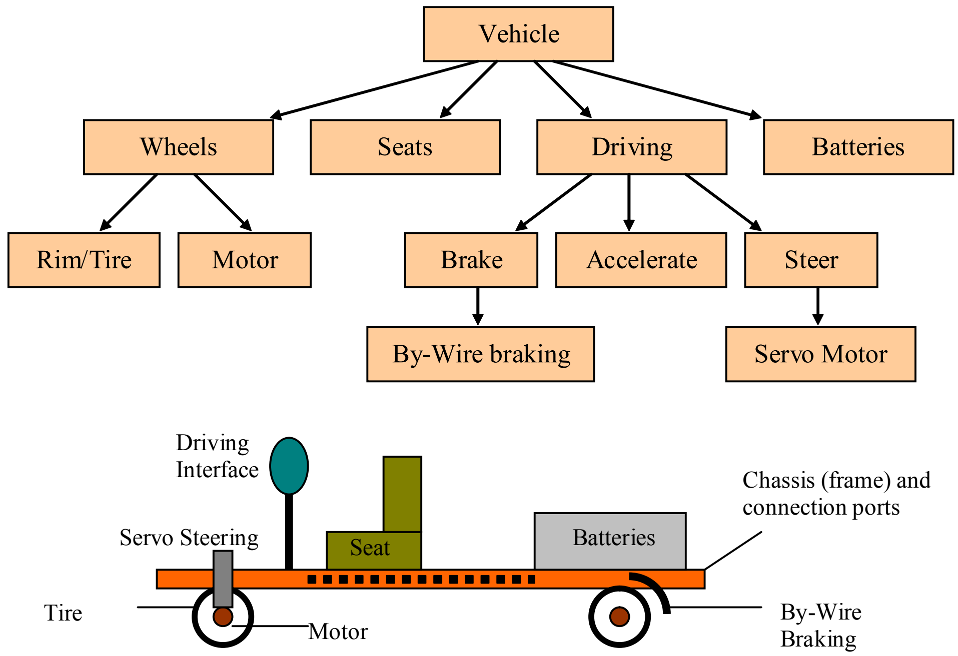


Figure 5. 21: One-to-one correspondence between the functional and physical structures of the proposed design.

This vehicle, unlike the conventional designs, consists of modules whose functions are directly relevant to driving, and whose purposes are obvious and understandable by the average user. These modules are: wheels for traction, seats for the occupants, control interfaces for driving, batteries for power, servo motors for steering, and brakes. Any of these independent modules can be utilized in other designs and configurations. For instance the same seat and batteries can be utilized in a car or in a wheelchair. Further, each module can be replaced by a different design for the same function, and without affecting the rest of the vehicle. For instance batteries can be replaced by fuel cells or by an engine/generator assembly; or the driving interface can be replaced by remote control. Various vehicles can thus be generated through the morphological combination of different solutions for modules as discussed in Chapter 4.

5.3. A Comparative Example

This section discusses two different designs for electric home and garden tools. The first design, which is chosen from the literature, demonstrates the process of designing a product for predetermined adaptations. The second design is developed for general adaptability without aiming at predetermined adaptations. The comparison of these two designs demonstrates the differences between *specific AD* and *general AD*.

5.3.1. A Versatile Home and Garden Tool

This example is chosen from [Gu 2003]. It is the design of a versatile home and garden tool that replaces a chainsaw, a hedge trimmer, and a grass edger. Figure 5.22 illustrates some commercially available models of these three products. This figure also highlights the functional similarities between them. These similarities indicate redundancy in having three individual products, hence suggesting the possibility of replacing these products with a single versatile design. The process of designing a versatile product for these functions is similar to the procedure described in Example 5.1. However, in this example the versatility of the product is achieved through a shared platform and several add-on modules.

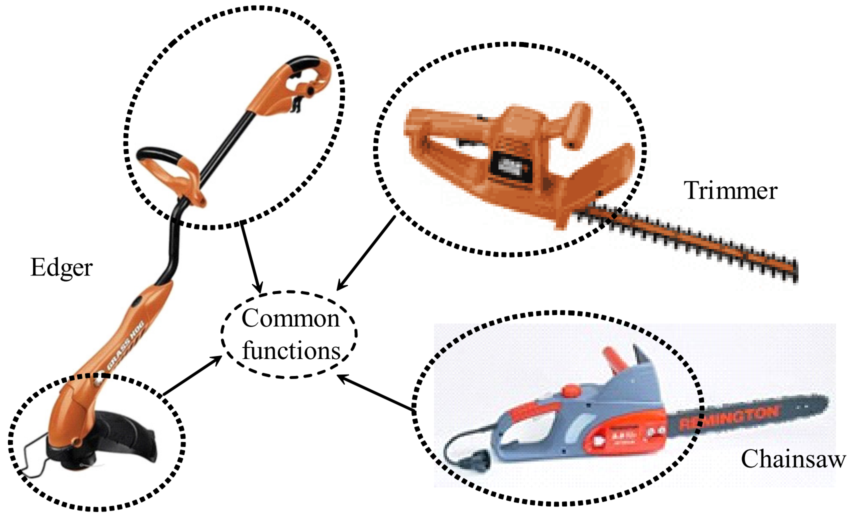


Figure 5.22: Common functions among chainsaws, trimmers, and edgers.

Step 1: Original FRs and Additional FRs

The functional requirements (FRs) for the individual products are:

Chainsaw:

- FR1: be held and controlled by hand.
- FR2: use electric motor to provide mechanical power.
- FR3: provide the rolling motion of the chainsaw.
- FR4: provide cutting action (the chainsaw).

Hedge Trimmer:

- FR1: be held and controlled by hand.
- FR2: use electric motor to provide mechanical power.
- FR3: provide the reciprocal motion of the trimmer.
- FR4: provide trimming action (the clippers).

Edger:

- FR1: be held and controlled by hand.
- FR2: use electric motor to provide mechanical power.
- FR3: provide the rotary motion of the edging blades (or strings).
- FR4: provide grass-cutting action (spinning blades or strings).

Figure 5.23 shows the functional structures of these three products and the conceptual solutions for these functions. The common elements among these designs are a handle and a switch for holding and controlling the device, and an electric motor for mechanical power. These common elements, shown on the left side of the figure, can be developed as a shared platform. The differentiating features shown on the right can be developed as add-on modules for individual applications.

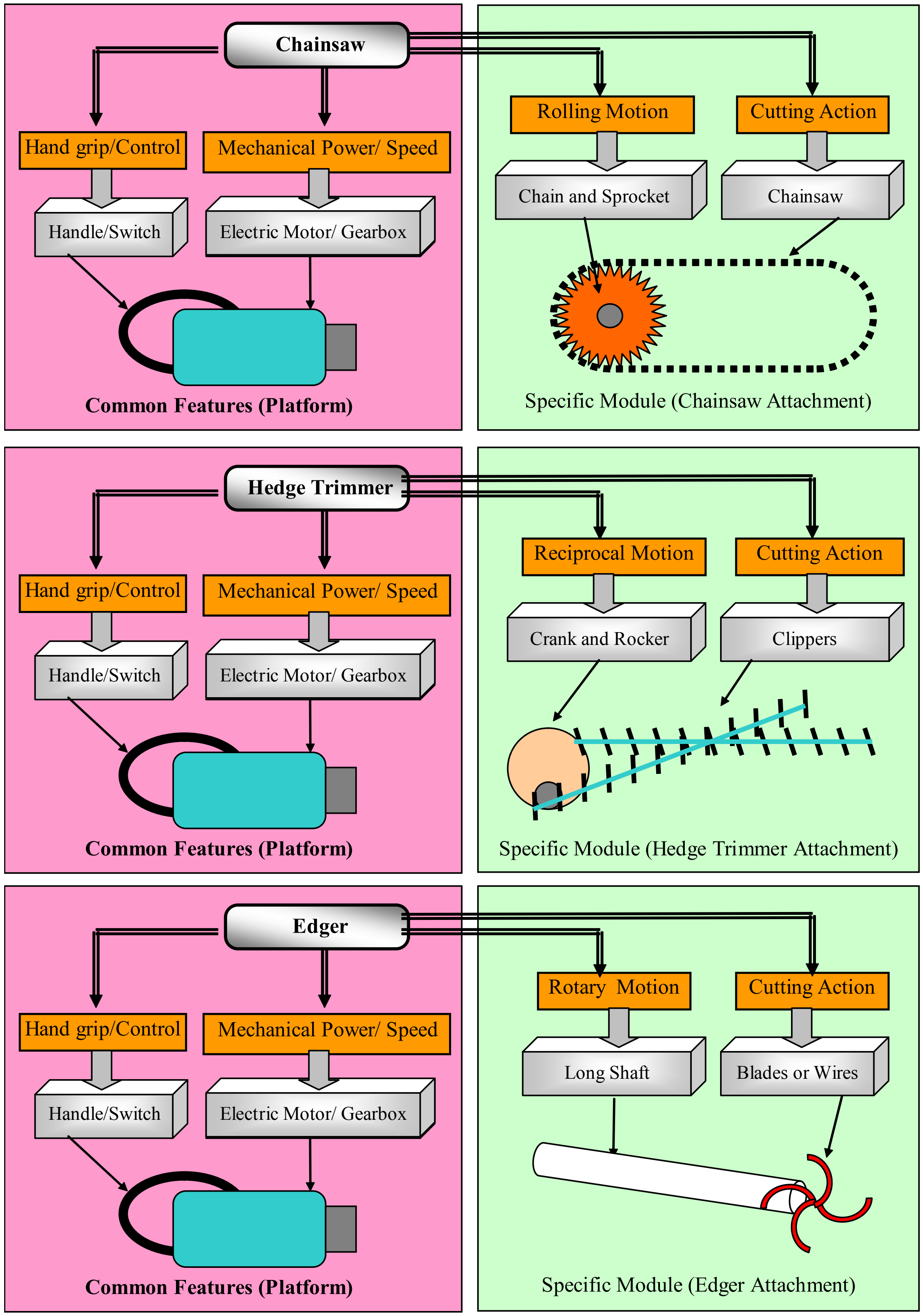


Figure 5.23: The functional structures of the chainsaw, the hedge trimmer, and the edger.

Step 2: Individual Designs

The conceptual designs for individual products are shown in Figure 5.23.

Step 3: The Conceptual Design of the Adaptable Product

Figure 5.24 illustrates the solid model of the overall adaptable design, developed by Gu and Slevinsky [Gu 2003]. The shared platform is shown on the left. It includes a motor, two handles, and a switch. This figure also displays the interface between the platform and modules. This interface is designed based on the guidelines of mechanical bus systems that incorporate lock-and-release and self-adjustment features [Gu 2003]. Three add-on modules for the three functions are shown in center of the figure, and the assembled chainsaw is shown on right.

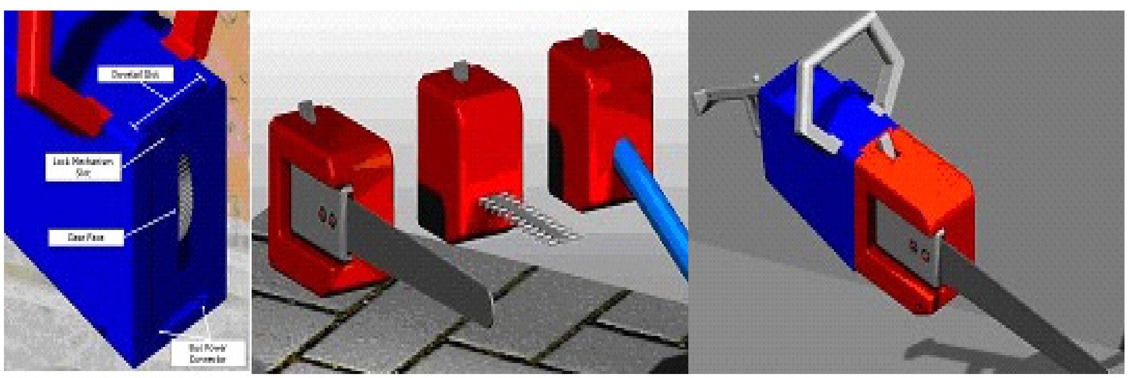


Figure 5.24: An adaptable design consisting of a platform, three modules, and proper interfaces.

Step 4: Adaptability

Since all three functions of this product are commonly used during its service life, the probability of every adaptation task is 1. In order to calculate the remaining parameters in the Equation (8) of Chapter 4, the *information content* of every task involved, including design, manufacturing, and adaptation, is measured by monetary value (Chapter 4, Section 4.1). The cost of every task is thus assumed to represent its 'total costs' in terms of the consumption of natural and artificial resources as discussed in Chapter 4. For instance it is assumed that the consumption of nature's absorption capacity is included in the price of artefacts. With these assumptions, the adaptability of this design can be calculated based on the costs of components and activities. The following costs are assumed here:

- The cost of an individual chainsaw: $80
- The cost of an individual hedge trimmer: $100
- The cost of an individual edger: $60
- The cost of the shared platform: $50
- The cost of the add-on module for the chainsaw: $50
- The cost of the add-on module for the hedge trimmer: $70
- The cost of the add-on module for the edger: $30
- The cost of swapping available modules: $0

The calculation of adaptability in Equation (8) is based on the saving which is achieved by adapting a product instead of replacing it. Therefore this calculation assumes a nominal operational mode (*initial state* S1) from which the product is adapted to another mode (*actual final state* AS2). For this example, the normal operational mode of this design is assumed to be the chainsaw. Also, this versatile product has been designed for effortless adaptation from one application to another. Therefore the cost of adaptation is in fact the initial extra investment for obtaining the add-on modules. The 'Adaptability Factor' (AF) for the three applications of this design can now be calculated using Equation (5) of Chapter 4:

$$AF(Spi) = \frac{Inf_{(ZERO \to IS2)} - Inf_{(S1 \to AS2)}}{Inf_{(ZERO \to IS2)}} = 1 - \frac{Inf_{(S1 \to AS2)}}{Inf_{(ZERO \to IS2)}}$$

<u>Chainsaw:</u> The adaptable chainsaw consists of a platform and a chainsaw module. Therefore its cost is \$50+\$50 = \$100. It is \$20 more than the cost of an individual chainsaw (\$80). This is the cost of adaptability for the nominal operational mode. Therefore the adaptability factor for the chainsaw is:

AF = 1 – (20/80) = 0.75

<u>Hedge Trimmer:</u> The cost of changing S1 to S2 is \$70, which is the price of the add-on module for the hedge trimmer. The $Inf_{(ZERO\ to\ IS2)}$ is \$100, which is the price of an individual hedge trimmer. Therefore:

AF = 1 - (70/100) = 0.3

<u>Edger:</u> The cost of changing S1 to AS2 is \$30, which is the cost of the add-on module. The $Inf_{(ZERO\ to\ IS2)}$ is \$60, which is the cost of an individual edger. Therefore:

AF = 1- (30/60) = 0.5

The total adaptability can be calculated as the sum of adaptability factors:

$$A(P) = \sum_{i=1}^{n} \Pr(Spi) AF(Spi)$$

Adaptability= 0.75 + 0.3 + 0.5 = 1.55

Since this number is more than 1, this adaptable design is justified.

5.3.2. The General AD of Home and Garden Tools

This section discusses the design of an electric chainsaw for general adaptability. In the previous example three applications for the versatile product were targeted during the design process; here the chainsaw is developed as an assembly of functional modules without targeting specific adaptations. The functional structure of a chainsaw (Figure 5.23) consists of an electric motor, handles and switches, and the chainsaw itself. For general adaptability, these functions should be accomplished by independent modules.

Figure 5.25 shows the design of the motor module. There are certain features in this design that make it more adaptable. First, the housing has regular shapes and surfaces, and also includes several mounting holes for possible future applications. Second, the two inputs of the motor, which are electric power and operation signals, are not designed as integral electric cords and integral control switches. Instead, they are designed as separate components because their requirements vary from one application to another. Therefore the housing has sockets so that the electric power and control switches can be connected in any configuration that suits a particular application. Third, the output is a rotating shaft which is more universally usable than other types of mechanical output. In particular, this output is more compatible with unforeseen applications than the gear output in the previous example. Fourth, the motor is a versatile type which can deliver different performance profiles for different applications. The electronic control systems and relay switches for this purpose are located inside the housing, and receive their operational signals through the electronic port (switch socket) on the motor.

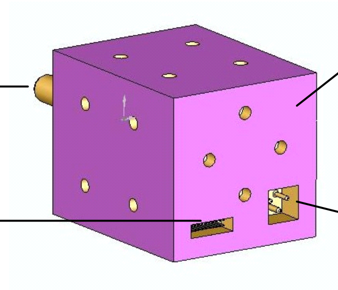


Figure 5. 25: An adaptable electric motor designed for the chainsaw.

Similar to the motor, the chainsaw should also be designed as an independent module. The input of this module is rotary mechanical power, and its output is the chainsaw action. Figure 5.26 illustrates the conceptual design of the chainsaw module. The input connection is made via the insertion of a shaft into the hub of a sprocket inside, which creates the rolling motion of the chain as the output. This module is also equipped with a handle, a protective shield, and switches for ON/OFF and speed control. These switches are connected to the motor through the connection socket, thus allowing this module to utilize different motors. It can be seen that this module does not depend on a specific electric motor for its operation, and can take its input from any source of mechanical power even from a gas engine. The flat surface on the side of the sprocket housing facilitates the installation of various sources of mechanical power.

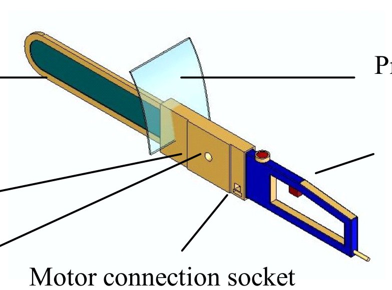


Figure 5. 26: The chainsaw module.

Figure 5.27 shows the chainsaw assembly which consists of the motor module, the chainsaw module, and a handle. This figure also shows several other power tools which utilize some of these functional modules. These applications were not known during the design of the chainsaw. Thus this example demonstrates the general adaptability of the proposed design to unforeseen adaptations.

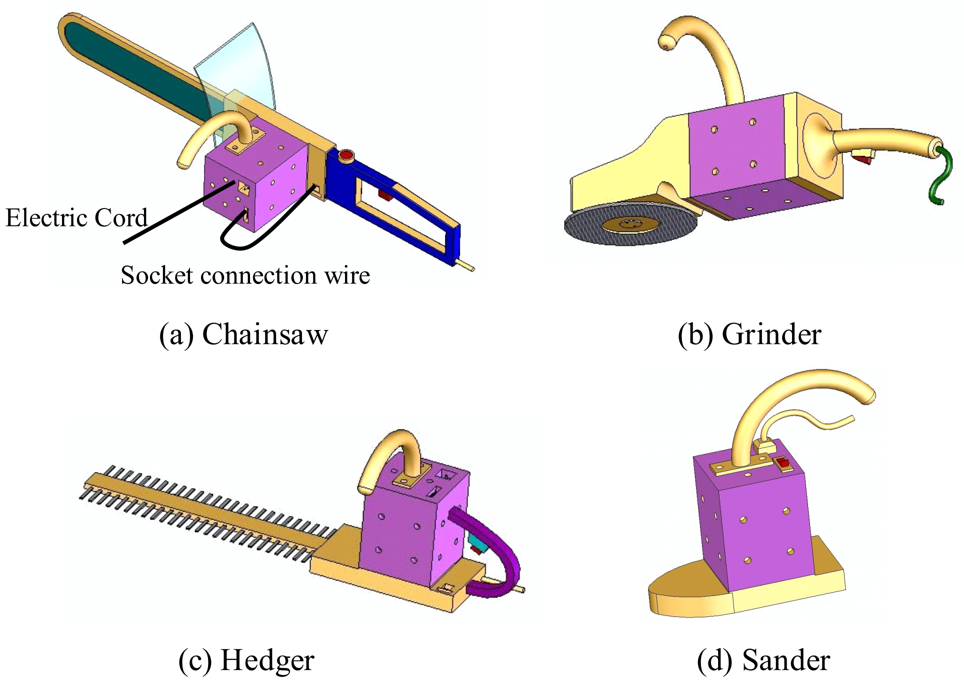


Figure 5. 27: Adaptable designs for power tools.

Discussion

This section discussed two different designs for power tools. The first design was prepared for predetermined adaptations, while the second design was developed for general adaptability. The first design performs well for its intended adaptations, but its platform and other modules are difficult to use in other applications. For instance the output of the platform is a gear, which requires its exact mating interface. The second design is not optimized for any particular adaptations, thus any adaptation task requires some effort. However, it can be adapted to unforeseen conditions with reasonable effort.

5.4. Calculation of Adaptability in General AD

In this chapter *adaptability* was only calculated for the examples of specific adaptability and not for the examples of design for general adaptability. Chapter 4 discussed that Sp, the set of *target adaptation tasks* considered in calculating the adaptability of product P, is a sub-set of the large set of all potential adaptations for P in its lifetime. If Sp is known at the time of design, the methodology of specific AD should be utilized and the adaptability can be calculated as the sum of *adaptability factors*. General AD, however, aims at developing a certain architecture for products that makes them more adaptable without targeting a specific set of adaptations. In this case it is not possible to evaluate the results according to the adaptability formula at the time of design. Instead, the adaptability of the final product for various adaptation tasks can be evaluated after adaptation tasks are decided. Once Sp is decided, the calculation of adaptability is similar to the procedure discussed in the examples of specific AD.

Chapter 6: Summary and Discussions

6.1. Thesis Summary

The goal of this thesis was to contribute to the development of a new design paradigm called *design for adaptability*, in which the *adaptability* of a product to varying service conditions is perceived as a design characteristic.

The *extension of utility* was identified as the underlying reason for adapting an artifact. This view revealed a logical relationship between the scarcity of resources and the need for AD. It also helped with the categorization of adaptation scenarios through considering different modes for extending utility. The study of literature within these categories revealed that current methods assume a predetermined set of adaptations, and then design a product for specific adaptabilities. This research thus focused on developing a design method for increasing the *general adaptability* of mechanical designs without targeting particular adaptations.

Modularization was recognized as a principal method for achieving adaptability. While predetermined adaptations related to *versatility*, *variety*, *upgrading*, and *customization* make up the segmentation criteria for the existing modular design methods, modularization for general adaptability required a different criterion. The study of the mechanical design process resulted in choosing *functions* as the segmentation criterion. This choice stemmed from the adaptability properties of a *rational functional structure*, and the proposition was that the subordination of the actual architecture of a product to a rational functional structure would yield similar adaptability in the design.

The architecture of such a product would be a hierarchy of *autonomous functional modules*, along with spatial *frames* and product-specific components when required. This architecture was called *frame-and-function*; the contrasts between this architecture and the popular *platform architecture* highlight the differences between *specific AD* and *general AD*. Three design etiquettes were proposed for achieving this architecture in the design of mechanical systems. First, designs should be modularized and relations between modules should be made as flexible as possible. Second, modules should perform meaningful functions which occur frequently in mechanical designs. This can be achieved through the proper choice of sub-FRs during the decomposition of problems. Third, modules should be developed as self-sufficient and independent systems. These autonomous modules can then be used in different configurations and in different systems without depending on a system's layout and other specifications.

The function-based division of subsystems is practiced in the design of virtual enterprises, software, electronic systems, and other areas of design in which system-level design and component-level design can be carried out independently. The inherent properties of mechanical design, however, hinder the application of this method in the design of mechanical systems. Two such properties were discussed in this thesis: the *structural connectivity* of mechanical components caused by spatial constraints; and the *functional ambiguity* of mechanical subsystems, which are typically designed for a specific function within their parent system and are not usable elsewhere. Although such inherent characteristics of mechanical design are somewhat inevitable, the thesis proposed several mitigating guidelines which help with making the design of mechanical systems closer to the ideal frame-and-function architecture.

The thesis also proposed a formula for assessing the adaptability of a design based on the amount of 'savings' which are achieved via adaptation. This assessment does not help with measuring the adaptability of a design based on its architecture. It is an after-the-fact method and requires information on costs and difficulties of performing adaptation tasks and procuring new products. These costs and difficulties are measured by *information content* in the proposed formula.

Then the thesis proposed a methodology for AD in which specific AD is performed first in order to take advantage of forecast information, and then general AD is performed in order to increase the adaptability of the design to unforeseen changes. The application of this methodology was demonstrated through several conceptual design examples. The highlights of the thesis are summarized in Table 6.1.

Issue	Approach	Solution	Conclusions/Results	Chapter
Justification of AD research	Identify the purpose of adaptation	*Extension of Utility*	Scarcity → service-based market → adaptation instead of new production	3
Types of adaptations?	Modes of extending utility	Criteria: time, subject, doer	(parallel/sequential; product/design; user/producer) → (*upgrading, variety, versatility, customization*)	3
Scope of AD	Set limits	3 limits	Avoiding terminology inconsistencies	1
Identify original research	Study current methods	*Specific AD, General AD*	General AD not established in mechanical engineering design.	1,2 ,3
General AD	Seek an ideal adaptable structure	Imitate the ideal *rational functional structure* in the physical structure	Function-based segmentation; autonomous functional modules assembled on spatial frames. (*frame and function* architecture)	4
Mechanical systems	Study inherent properties	Find ways to avoid constraints	Guidelines	3, 4
Methodology	Combination	Specific AD prior to general AD	The overall AD methodology	4
Assessment	Measuring adaptability	Criterion is *saving*	Trade-off formula applicable to specific AD only.	4

Table 6. 1: Highlights and organization of the thesis.

6.2. Discussions

This section discusses three issues related to this research. First, the concept of "function" in function-based modularization for general adaptability will be discussed to show that the distinction between specific AD and general AD is related to the "scope of applicability" of mechanical subsystems and their functions. Second, the measurement and minimization of "information content" and the inclusion of "quality" in the assessment of a design will be discussed. Third, the circumstances that justify the higher costs of making products more adaptable are discussed using the proposed formula for measuring adaptability. These discussions will be utilized to draw the conclusions of this research in the next chapter.

6.2.1. Function-Based Modularization

Mechanical systems are naturally constructed from subsystems or modules. Methods of *modular design* further increase the modularity of mechanical systems. This thesis proposed *function-based modularization* as the primary method for increasing the general adaptability of mechanical designs. The distinction between this method and the conventional methods of developing modular products is the emphasis which is given to *functions*: while the division of subsystems in conventional design is driven by problem-specific criteria which might not be related to functions, in 'general AD' the division of subsystems exclusively aims at developing modules which perform meaningful functions. Two implications of this distinction are discussed in this section.

Function as Division Criterion

In 'general AD', FRs at every level of decomposition are chosen in such a way that each physical FR represents a useful function. Then for each FR an autonomous module is developed which is self-sufficient and is capable of delivering its function independently from the larger context it is placed in. These two etiquettes, the usefulness of functions and the autonomy of their corresponding physical modules, determine the modularization scenario in the architecture of a product. Thus the division of subsystems is based on the meaning, recurrence, or usefulness of their functions. This approach avoids the four shortcomings of conventional clustering methods discussed in Chapter 2: combinatorial complexity does not exist because only a few rational options have to be considered; uncertain numerical data are not utilized in data-sensitive algorithms; functional ambiguity of modules is reduced because modules are explicitly designed for meaningful and recurring functions; and the designer's freedom in changing the design of a module can be maintained because modules are autonomous and their design/modification can be carried out independently.

A Function's *Meaning*

The above discussion assumed that some modules perform a more meaningful function than others. The fact is that there is no general agreement on the meaning of functions in mechanical engineering design. In *Systematic Design* the function of a component is defined as its effects on materials, energies, and signals [Pahl 1988] [1]. This narrative description does not reflect the purpose of a component and the rationale behind its existence. In *Axiomatic Design*, on the other hand, functions are defined as the necessary and sufficient description of *goals* [Suh 1990]. This definition does not reflect the actual physical effects thus both physical and non-physical requirements are called FRs in Axiomatic Design.

The opinions on how to 'represent' functions are even more diverse [Hashemian 1997-a]. Of relevance to this discussion is the requirement for the functions to be represented in *solution-neutral* terms. Appendix 1 presents a function representation scheme in which a function is described by its actions on physical entities, where *actions* and *entities* are chosen from predetermined taxonomies. Despite their academic research value, such solution-neutral representation methods have not proven useful to the design

[1] In *Systematic Design* the function of a component is defined as what it actually does; the functional structure of a product is obtained by the replacement of every component with its function. By this popular definition of a functional structure, the "subordination of physical structure to functional structure" would be a redundant statement. Therefore, general AD emphasizes that the physical structure should be subordinated to a *rational functional structure* defined in Chapter 4. This means that relations between subsystems should be *causal*. Few mechanical systems follow this structure in their designs.

practice in industries. The reason is that solution-neutrality for the meaningful representation of functions is a *relative* parameter. It depends on the level of design hierarchy to which the task at hand belongs. People, industries, companies, and design teams use different vocabularies for describing functions. For a designer who has been assigned with the task of designing a rack-and-pinion, the name of the device might be an acceptable description of FRs. In a larger context, for the average user of a vehicle "steering" is a meaningful description of a function while "rack-and-pinion" refers to a particular mechanism. Therefore, solution-neutrality, meaning, and usefulness of a function are relative parameters which depend on the scope of a module's application. More "generality" in this context means: the design task is at a higher level in the design hierarchy; the function is applicable to a broader range of applications; the function is of interest to a larger audience; and so on.

The beginning of this section emphasized that the distinguishing feature of *function-based segmentation* is the development of modules which perform "useful" functions. The above discussion revealed that this usefulness is a relative parameter which reflects the "generality" of a module's function. Therefore, the main difference between a *functional module* developed by *general AD* and a *product-specific module* developed by *specific AD* is in fact in their scope of applicability. That is, general AD aims to increase the scope of applicability of mechanical systems. A similar statement can be made on the difference between *platform architecture* and *frame-and-function architecture*: while a platform's function is usable within a portfolio of products, the 'functional modules' of a frame-and-function architecture perform functions which are usable in a broader spectrum of applications.

6.2.2. Information Content

Chapter 4 defined the *information content* (IC) of a design as the 'total costs' of materializing that design. These costs reflect the consumption of resources of all kinds including natural resources such as materials, energy, pollution absorption capacity, land, water, and so on; and artificial resources such as industrial infrastructure, subcontractors, labor and expertise, software and hardware, and so on. This interpretation of information content is somewhat different from the definition of IC in Axiomatic Design, which relates IC to the "probability of success" and measures it by "bits" [Suh 1990]. Three distinguishing aspects of this interpretation are discussed in this section: the use of monetary value for measuring IC, the encouragement of a minimalist view leading to such principles as "self-help" in design, and the inclusion of "quality" in evaluating a design.

Measuring IC

The consumption of resources is most commonly measured by monetary means. For instance, the consumption of Nature's absorption capacity can be measured by financial criteria; this valuation of "pollution" might include penalties and fines, loss of customers due to adverse publicity, class action suits, and other consequences of causing environmental impacts [Tipnis 1998]. Monetary means are also used for the valuation of rare natural resources. For instance, the value of 200 million tons of topsoil blown off the U.S. Great Plains in one 1934 dust storm has been estimated at $9 trillion [NGM 2001]. The measurement of resources by financial means, however, requires a fair and intelligent valuation and *pricing* system.

Minimizing IC

The IC of a design depends on how much the existing states have to be altered in order for that design to materialize successfully. Therefore, the minimization of information content means that a good design should require minimal alteration of the existing states. This philosophy leads to the development of design principles such as the *self-help* principle discussed by Pahl and Beitz [Pahl 1988].

Quality

The success of a design can rarely be decided in a binary fashion: successful or unsuccessful. Success can be achieved at various levels. The level of success represents the '*quality'* of a design. Quality is an indication of how well the initial objectives are fulfilled. Therefore, an evaluation schema should include the parameter of quality in addition to the main parameter of IC. This is shown in Figure 6.1. In this figure, the goal of design is to set the value of a functional requirement (FR on the horizontal axis) within an acceptable range. Although every design which delivers the FR within this range is considered 'acceptable' or 'successful', different values within this range yield different values for quality. In this figure quality as a function of FR is shown on the horizontal plane (XY plane). In a typical design situation, the final value of FRs can be determined only probabilistically because of manufacturing variations and other noise factors [Suh 1990]. In this figure the probability density function of FR is shown on the vertical plane (XZ plane). Therefore, the level of success, measured by quality, can be calculated by a dual integral which corresponds to the enclosed volume in the figure.

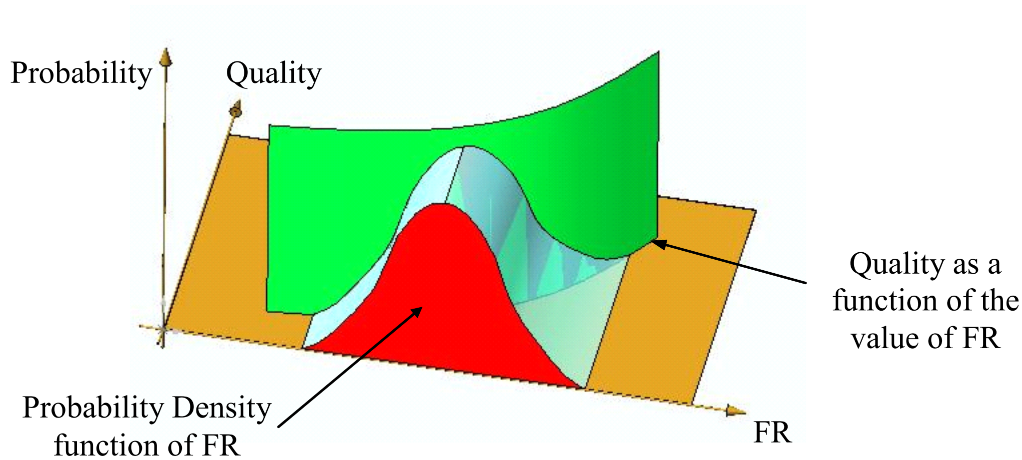


Figure 6. 1: Both quality and probability should be used in the evaluation of a design.

6.2.3. Justification of Design for Adaptability

The formula for measuring adaptability resulted in several rules and guidelines in Chapter 4. Particularly, it was stated that a negative value for the adaptability factor (AF) means that the adaptation task is not justifiable. When the procurement of a new product is difficult due to the scarcity of resources, AF values will change and can justify adaptability. This can be discussed in the context of an example.

Assume a product costs \$100; it can be adapted to perform the function of another product which costs \$80 and the cost of adaptation is \$500; the cost of an adaptable design of the same product is \$200, and the cost of adapting this design is \$0. For the original product, the AF for the adaptation task is:

AF (original) = (80-500)/80 = -5.25 (not justified!)

For the adaptable design of the product, AF for the adaptation task is:

AF (adaptable) = [(80-0) - (200-100)] / 80 = -0.25 (not justified!)

Now assume these products have to be shipped to a far location for an expedition (e.g. a space expedition). The transportation cost for each product is $400. This has to be added to the cost of products, reminding that the parameter in the formula is IC which represents *total costs*. Thus:

AF (original) = (480-500)/480 = -0.04 (not justified!)

AF (adaptable) = [(480-0) - (200-100)] / 480 = 0.8 (justified!)

This example shows that the scarcity of resources justifies the higher costs of adaptable design and the costs and difficulties of performing adaptation tasks.

Chapter 7: Conclusion

This chapter lists the conclusions and contributions of this research and briefly discusses future work related to the subject.

7.1. Conclusions of This Research

Importance of AD

The potential benefits of AD encompass the *environment*, the *user*, and the *producer*. Its environmental benefits are achieved through creating new usage for idle or obsolete products, thus postponing product retirement and reducing product replacement. Its customer benefits stem from the customizability and adaptability of products to varying needs. Its producer benefits result from the ability to reuse proven designs and production processes instead of creating new ones. Increasing environmental concerns, rising customer expectations, and increasing competition among producers motivate research and development of AD as an emerging paradigm for mechanical design.

The AD Framework

This thesis established a framework for AD as a new design paradigm based on a broad definition of adaptability, which includes user adaptations and producer adaptations, design adaptations and product adaptations, and adaptations which occur over the course of time and those which are unrelated to time. The setting up of this framework involved original classifications and definitions, and the study of circumstances and parameters which influence adaptability-related decisions during design. In the proposed framework, existing design methods can be categorized as "targeting predetermined adaptabilities" during the design process.

Design for General Adaptability:

Function was identified as a general modularization criterion which can be considered in addition to, or in the absence of, predetermined adaptability targets.

The fundamental arguments for utilizing functions in designing for general adaptability are supported by the importance of functions in a service-based market on one hand, and by the adaptability properties of "rational functional structures" on the other hand.

For the actual implementation of this idea in mechanical design, this thesis proposed the construction of systems from autonomous functional units in a "frame-and-function architecture". This requires the overcoming of the inherent difficulties of mechanical design, which were identified as *structural connectivity* and *functional ambiguity*. For this purpose, guidelines and methods were developed.

In conclusion, the proposition for the development of adaptable products is supported by logical reasoning, while the proposed methods and guidelines need to be tested through practical design problems for further development and improvement.

The Overall Methodology

The AD methodology combines *specific AD* and *general AD* in a sequential manner. As discussed in Chapter 6, the distinction between these two, similar to the distinction between the *platform architecture* and the *frame-and-function architecture*, leads to a fundamental discussion on the “meaning” of a function. It can be concluded that the premise of the overall methodology, including both specific AD and general AD, is to generalize the applicability of designs and systems via making their functions less context-dependent and more universally usable. This conclusion also reveals the relation between a function’s meaning and its generality of usage. This finding is helpful for the development of function representation schemes such as the one discussed in Appendix 1.

Measuring Adaptability

The proposed formula for measuring adaptability is directly based on the rationale of decision making and results in a tangible quantity for adaptability. The methods of evaluating and quantifying the elements of this formula, however, need to be further developed. In particular, the relation between *information content* and *total costs* requires extensive research in order to develop practical quantification schemes.

7.2. Contributions

- Treating *adaptability* as a design characteristic similar to manufacturability, thus setting grounds for a new design paradigm.
- Identifying the *extension of utility* as the purpose of adaptation, thus establishing a relationship between the scarcity of resources and the need for AD.
- Developing a categorization of adaptabilities in engineering design, which led to the conclusion that current methods are concerned with design for *specific adaptabilities*, while there is no formal method for the design of mechanical systems for *general adaptability*.

- Developing a proposition that the general adaptability of a product can be increased through the subordination of its physical structure to an ideal *rational functional structure*.
- Proposing the *frame-and-function architecture* for the general adaptable design of mechanical systems. A system with this architecture consists of several *autonomous functional modules* which correspond to a rational functional structure and a spatial *frame* for the assembly of these modules according to the embodiment requirements of a design problem.
- Proposing methodology and guidelines for *design for adaptability*.
- Introducing a novel interpretation for the concept of *information content*.
- Developing a measure for adaptability based on the amount of saving which is achieved by adaptation.

7.3. Future Work

The calculation of adaptability in this thesis is based on comparing the information content of an adaptation task with the information content of producing a new product. Therefore this method is applicable only when the adaptation task is known and $Inf_{(S1 \rightarrow S2)}$ can be measured. An alternative approach is to calculate the adaptability of a product based on its design characteristics. The design characteristics that affect the adaptability of a product include: the correspondence between the functional and the physical structures, the ratio between the number of functional components and the number of product-specific components, the difficulty (information content) of assembly and disassembly tasks, the flexibility and compatibility of interfaces among modules, and the utilization of smaller sizes and generic forms. Developing methods for the quantification of these characteristics for the calculation of adaptability would be an extension of the current research.

While Axiomatic Design defines IC by the ‘probability of success’, this thesis defines it as the ‘cost of success’. The probability of success, however, might be an indication of costs. That is, a design that has a higher probability of success usually costs less to succeed than a design with a lower probability of success. Since probability of success is relevant only when design is incomplete, it might be possible to use it as a heuristic rule for assessing the ‘cost of success’ of solution alternatives generated for FRs during the design process. The study of the relation between probability and cost might lead to the validation of the above heuristic rule.

The methods proposed in this thesis should be tested through more concrete case studies, especially with respect to cost and feasibility.

References

[Albano 1993] Albano, L. D., Suh, N. P., "Information axiom and its implications, Proceedings of the 1993 ASME Winter Annual Meeting, American Society of Mechanical Engineers, Design Engineering Division, V. 66, 1993, pp 1-11.

[Allen 1962] Allen M. S. “Morphological Creativity”, Prentice Hall, Englewood Cliffs, NJ, 1962.

[Altshuller 1984] Altshuller, G., “Creativity as an Exact Science, The Theory of the Solution of Inventive Problems (TSIP)”, translated from Russian, Gordon and Breach Pub. 1984.

[Altshuller 1998] Altshuller, G., "40 Principles: TRIZ Keys to Technical Innovation", 1st edition, Technical Innovation Center Inc., USA, 1998.

[Altshuller 1999] Altshuller, G., "The Innovation Algorithm" (Translated by L. Shulyak, S. Rodman), Technical Innovation Centre Inc., USA, 1999.

[Ayres 1998] Ayres, R. U., “Toward a zero-emissions economy”, Environmental Science and Technology, V. 32, No. 15, August 1, 1998, pp 366-367.

[Bechler 1997] Bechler, B., “Evolution of the virtual enterprise”, Annual International Conference Proceedings - American Production and Inventory Control Society, 1997, pp 65-67.

[Belhe 1996] Belhe, U., Kusiak, A., "House of quality in a design process", International Journal of Production Research, V.34, No. 8, 1996, pp 2119-2131.

[Bhaskar 1995] Bhaskar, V., and Glyn A. (eds). “The North, the South and the Environment”, Earthscan Publications and United Nations University Press., London, 1995.

[Boothroyd 1983] Boothroyd, G., Dewhurst, P., “Design for Assembly: A Designers’s Handbook”, Boothroyd and Dewhurst Inc., Wakefield, Rhode Island, 1983.

[Boothroyd 1994] Boothroyd, G., "Product design for manufacture and assembly", Computer Aided Design, V. 26, No. 7, 1994, pp 505-520.

[Brezet 2001] Brezet, J.C., A.S. Bijma, J. Ehrenfeld, and S. Silvester, "The Design of Eco-Efficient Services: Method, Tools And Review of the Case Study Based ‘Designing Eco-Efficient Services’ Project", Design for Sustainability Program, Delft University of Technology, Delft, The Netherlands,2001.

[Broadbent 1981] Broadbent, G., 1981, "The Morality of Designing", Design: Science: Method, Proceedings of The 1980 Design Research Society Conference," pp. 309-328, Westbury House, Guilford, England.

[Byrne 1993] Byrne, J. G., Barlow, T."Structured brainstorming: A method for collecting user requirements", Proceedings of the Human Factors and Ergonomics Society, V. 1, 1993, pp 427-431.

[Chen 1994] Chen, Wei, Rosen, David, Allen, Janet K., Mistree, Farrokh, "Modularity and the independence of functional requirements in designing complex systems", American Society of Mechanical Engineers, Design Engineering Division (Publication)

DE, Concurrent Product Design, Proceedings of the 1994 International Mechanical Engineering Congress and Exposition, V. 74, 1994, pp 31-38.

[Chen 2002] Chen, Li-Chieh, Lin, Li, "Optimization of product configuration design using functional requirements and constraints", Research in Engineering Design, V. 13, No. 4, 2002, pp 167-182.

[Cherian 2001] Cherian, R.P., Midha, P.S., Smith, L.N., Pipe, A.G., "Knowledge based and adaptive computational techniques for concurrent design of powder metallurgy parts", Advances in Engineering Software, V. 32, No. 6, 2001, pp 455-465.

[Clausing 1994] Clausing D.P, Total Quality Development, ASME Press, New York, 1994.

[Cohen 1992] Cohen, R., Lipton, M.G., Dai, M.Q., Benhabib, B., "Conceptual design of a modular robot", Journal of Mechanical Design - Transactions of the ASME, V. 114, No. 1, 1992, pp 117-125.

[Condoor 1992] Condoor, S., Shankar, R., Brock, R., Burger, P., Jansson, G., "Cognitive framework for the design process", American Society of Mechanical Engineers, Design Engineering Division, V. 42, 1992, pp 277-281.

[Condoor 1999] Condoor, S. S., Weber, R. G., “Model for conceptual design methodology”, American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 103, 1999, pp 57-64.

[Coyne 1987] Coyne, R. D., Rosenman, M. A., Radford, A. D., and Gero, J. S. “Innovation and creativity in knowledge-based CAD, Expert Systems in Computer-Aided Design”, ed. J. S. Gero, Amsterdam, North-Holland, pp. 435-465. 1987.

[Coyne 1990] Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., and Gero, J. S. “Knowledge-Based Design Systems”, Reading: Addison-Wesley. 1990.

[Cross 1980] Cross, N. J. and Walker, D., "Design Method and Scientific Method", Design: Science: Method, Wcstbury House, Guilford. England, 1980, pp. 18-27.

[Dahmus 2001] Dahmus, J.B., Gonzalez-Zugasti, J.P., Otto, K.N., "Modular product architecture", Design Studies, V. 22, No. 5, 2001, pp 409-424.

[de Bono 1992] de Bono E., “Serious Creativity”, Harper Collins Publishers, New York, 1992.

[DeLit 2003] De Lit, P., Delchambre, A., "Integrated design of a product family and its assembly system", Kluwer Academic Publishers, Boston, 2003.

[Dixon 1988] Dixon. J.R., "On Research Methodology Towards A Scientific Theory of Engineering Design," Artificial Inelegance for Engineering. Design, Analysis and Manufacturing (AI-EDAM). Vol. 1, No. 3, 1988.

[Domb 1998] Domb, E., Kowalick, J., "Applying TRIZ to develop new designs for the future of the airbag", SAE Special Publications, Airbag Technology, V. 1333, 1998, pp 65-71.

[Doriben 2003] Doriben, H. T., Durkopp, K., "Mechatronics and drive-by-wire systems advanced non-contacting position sensors", Science Direct, Control Engineering Practice, No. 11, 2003, pp 191-197.

[Dorst 2001] Dorst, K., Cross, N., "Creativity in the design process: Co-evolution of problem-solution", Design Studies, V. 22, No. 5, 2001, pp 425-437.

[Du 2001] Du, X., Jiao, J., Tseng, M., "Architecture of product family: Fundamentals and methodology", Concurrent Engineering Research and Applications, V. 9, No. 4, 2001, pp 309-325.

[Eppinger 1990] Eppinger, Steven D., Whitney, Daniel E., Smith, Robert P., Gebala, David A.,"Organizing the tasks in complex design projects", American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 27, Design Theory and Methodology - DTM '90, 1990, pp 39-46.

[Farrell 2001] Farrell, R. S., Simpson, T. W., "Improving commonality in custom products using product platforms", 2001 ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference, V.2, Sep 9-12 2001, pp 1075-1085.

[Finger 1989] Finger, S. and J. R. Dixon (1989). "A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes," Research in Engineering Design, Vol. 1, pp. 51-67, Springer-Verlag, New York, NY, USA.

[Fujimoto 2003] Fujimoto, J., Umeda, Y., Tamura, T., Tomiyama, T., Kimura, F., "Development of Service-Oriented Products Based on the Inverse Manufacturing Concept", Environmental Science and Technology, V. 37, No. 23, 2003, pp 5398-5406.

[Gero 1985] Gero. J.S. and Coyne, R.D., "Knowledge-Based Planning as a Design Paradigm," Design Theory in Computer-Aided Design. Proceedings of the IFIP WG 5.2 Working Conference 1985. Tokyo. Yoshikawa, H. and Warman, E.A., eds., North Holland, Amsterdam, 1985, pp. 261-295.

[Gero 1989a] Gero, J. S. (1989), “Prototypes: a basis for knowledge-based design”, in J.

S. Gero and T. Oksala (eds.), Knowledge-Based Systems in Architecture, Acta Scandinavica, Helsinki, pp. 11-17.

[Gero 1989b] Gero, J. S. (1989), “Knowledge-based computer-aided design”, in F. Kimura and A. Rolstadas (eds.), Computer Applications in Production and Engineering, North-Holland, Amsterdam, pp. 13-20.

[Goldman 1995] Goldman, S., Nagel, R., Preiss, K., “Agile Competitors and Virtual Organizations- Strategies for Enriching the Customer”, Van Nostrand Reinhold, NY, 1995.

[Gonzalez-Zugasti 2000] Gonzalez-Zugasti, J.P., Otto, K. N., Baker, J. D., "Method for architecting product platforms", Research in Engineering Design - Theory, Applications, and Concurrent Engineering, V. 12, No. 2, 2000, pp 61-72.

[Gou 2003] Gou, H., Huang, B., Liu, W., Li, X., “A framework for virtual enterprise operation management”, Computers in Industry, V. 50, No 3, 2003, pp 333-352.

[Gu 1997] Gu, P. , Hashemian, M., Sosale, S. , "Integrated modular design methodology for life-cycle engineering", CIRP Annals - Manufacturing Technology, V. 46, No. 1, 1997, pp 71-74.

[Gu 1999] Gu, P., Sosale, S., "Product modularization for life cycle engineering", Robotics and Computer-Integrated Manufacturing, V. 15, No. 5, 1999, pp 387-401.

[Gu 2002] Gu, P., “Adaptable Design Using Bus Systems”, Proceedings of the 10th International Manufacturing Conference in China (IMCC2002), Xiamen, China, Oct. 11-12, 2002.

[Gu 2003] Gu, P. and Slevinsky, M., “Mechanical Bus for Modular Product Design,” Annals of CIRP, Vol. 52, No. 1, 2003, pp. 113-116.

[Gu 2004] Gu, P., Hashemian, M., Nee A., “Adaptable Design”, CIRP Annals - The Proceedings of the CIRP General Assembly, Poland, 2004.

[Hashemian 1995] Hashemian M., “Assisting Conceptual Mechanical Design”, Masters thesis, University of Calgary, 1995.

[Hashemian 1996] Hashemian, M., and Gu, P., "Representation and Retrieval of Design Knowledge for Conceptual Mechanical Design", Proceedings of the 3rd CIRP Workshop on Design and Implementation of Intelligent Manufacturing Systems, June 19-22, 1996, Tokyo, Japan, pp. 85-94.

[Hashemian 1997-a] Hashemian, M., Gu, P., “A function representation scheme for conceptual mechanical design”, Proceedings of the 11th International Conference on Engineering Design (ICED 97), V. 2, 1997, pp 65-71.

[Hashemian 1997-b] Hashemian, M., Gu, P., "Modeling the Creative Knowledge of Conceptual Design", Proceedings of the 11th International Conference on Engineering Design (ICED 97), V. 2, 1997, pp 211-214.

[Hashemian 2004] Hashemian, M., Gu, P., "Postponing Retirement in Engineering Design”, Proceedings of the first CDEN conference, Montreal, 2004.

[Hauschild 1998] Hauschild, M., Wenzel, H., “Environmental Assessment of Products”, Chapman & Hall Publishers, 1998.

[Hazelrigg 1996] Hazelrigg, G.A., “Systems Engineering, An Approach to Information-Based Design”, Prentice Hall, NJ, 1996.

[Hazelrigg 1998] Hazelrigg, G.A., “Framework for decision-based engineering design”, Journal of Mechanical Design, Transactions of the ASME, V. 120, No. 4, 1998, pp 653-658.

[Higgins 1996] Higgins, James M., "Innovate or Evaporate: Creative Techniques for Strategists', Long Range Planning, V. 29, No. 3, 1996, pp 370-380.

[Hillstrom 1994] Hillstrom, F., “Applying axiomatic design to interface analysis in modular product development”, American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 69-2, 1994, pp 363-371.

[Hubka 1980] Hubka V., “Principles of Engineering Design”, Butterworth Scientific, London, 1980.

[Hubka 1988] Hubka, V., Eder E. W., "Theory of technical systems : a total concept theory for engineering design", Springer, Berlin, 1988.

[Hubka 1992] Hubka V., Eder W.E., “Engineering Design”, Heurista, Zurich, Switzerland, 1992.

[Hubka 1996] Hubka, V., Eder, W. E., "Design science : introduction to needs, scope and organization of engineering design knowledge", Springer Verlag, London, 1996.

[Hykin 1975] Hykin, D.H.W. and Lansing, L.C., "Design Case Histories: Report of a Field Study of Design in the United Kingdom Engineering Industry," Proceedings of the Institute for Mechanical Engineers, Vol. 189, No. 23, 1975.

[Ishii 1994] Ishii, K., Eubanks, C.F., DiMarco, P., “Design for Product Retirement and Material Life Cycle”, Materials and Design, V. 15, No. 4, pp 225-233.

[Ishii 1995] Ishii, K., Juengel, C., Eubanks, C.F., “Design for Product Variety: Key to Product Line Structuring”, ASME Design Technical Conference Proceedings, 1995, V.

2, pp 499-506.

[Ishii 1998] Ishii K., “Modularity: A key concept in Product Life-cycle Engineering”, Stanford University, Stanford, 1998.

[Iyengar 1994] Iyengar, G., Lee, Chun-Liang, Kota, S., "Towards an objective evaluation of alternate designs", Journal of Mechanical Design, Transactions Of the ASME, V. 116, No. 2, 1994, pp 487-492.

[Jeong 2002] Jeong, I., Leon, V. J., "Decision-making and cooperative interaction via coupling agents in organizationally distributed systems", IIE Transactions (Institute of Industrial Engineers), V. 34, No. 9, 2002, pp 789-802.

[Kamrani 2003] Kamrani, A. K., Gonzalez, R., "A genetic algorithm-based solution methodology for modular design", Journal of Intelligent Manufacturing, V.14, No. 6, 2003, pp 599-616.

[Koestler 1967] Koestler, A., “The Ghost in the Machine”, Hutchinson Publishers, 1967.

[Kolodner 1983] Janet Kolodner, “Maintaining Organization in a Dynamic Long-term Memory”, Cognitive Science, Vol. 7, (1983), pp. 243-280.

[Kolodner 1992] J. L. Kolodner, "CBR for Design", in Pu, Faltings, Maher, Navinchandran, Oxman, (eds.), "Case-Based Design Systems", Workshop Proceedings, AID'92 Second International Conference on Artificial Intelligence in Design, Carnegie Mellon University, Pittsburgh, Penn., USA, 1992.

[Kolodner 1993] J. L. Kolodner, “Case-based Reasoning”, Morgan Kaufmann, 1993.

[Kourmaev 2003] Kourmaev, R., Teplitsky, A., "Application of inventing problem-solving theory (TRIZ) in pipeline technologies", New Pipeline Technologies, Security, and Safety, V. 2, 2003, pp 1465-1471.

[Kushnir 2003] Kushnir, E., Sheehan, T., "Development of machine tool structure at the early stages of design process", American Society of Mechanical Engineers, Pressure Vessels and Piping Division (Publication) PVP, V. 470, 2003, pp 121-127.

[Lee 2003] Lee, B., Saitou, K., "Assembly synthesis with subassebly partitioning for optimal in-process dimensional adjustability", Proceedings of the ASME Design Engineering Technical Conference, V. 2A, 2003, pp257-268.

[Li 1998] Li H., Wu X., "learning design knowledge as generalization", in Cybernetics and Systems, V. 29, 1988, pp. 181-207.

[Li and Zhang 1998] Li, Q., Zhang, W.J., "Computer comparison of design knowledge", Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, V. 212, No. B8, 1998, pp 635-645.

[Lindahl 2001] Lindahl, M., Ölundh, G., "The Meaning of Functional Sales," in de Ron, A.J. (ed.): Proceedings of the 8th CIRP International Seminar on Life Cycle Engineering – Life Cycle Engineering: Challenges and Opportunities, June 18-20, 2001, Scientific–Technical Union of Mechanical Engineering of Bulgaria, 2001, pp. 211-220.

[Lu 2003] Lu, B., Gu, P., "Systematic Life Cycle Design for Sustainable Product Development", ASME 2003 design engineering technical conference, September 2003.

[Luce 1957] Luce, R. D. and H. Raiffa, "Games and Decisions", John Wiley, NY, 1957.

[Maher 1987] Maher, M.L., "A Knowledge-Based Approach to Preliminary Design Synthesis," Engineering Design Research Center Paper EDRC-12-14-87, Carnegie

Mellon University, 1987.

[Maher 2003] Maher, M. L., Tang, H., "Co-evolution as a computational and cognitive model of design", Research in Engineering Design, V. 14, No. 1, 2003, pp 47-63.

[Mann 2002] Mann, D., "Manufacturing technology evolution trends", Integrated Manufacturing Systems, V. 13, No. 2, 2002, pp 86-90.

[Markeset 2001] Markeset, T., Kumar, U. , “R and M and risk-analysis tools in product design, to reduce life-cycle cost and improve attractiveness”, Proceedings of the Annual Reliability and Maintainability Symposium, 2001, pp 116-122.

[Marshall 1999] Marshall, R., Leaney, P.G., Botterell, P., "Modular design", Manufacturing Engineer, V. 78, No. 3, Jun 1999, pp 113-116.

[Martin 2002] Martin M. V., Ishii K., “Design for variety: developing standardized and modularized product platform architectures, Research in Engineering Design, V. 13, No. 4, 2002, pp213 - 235.

[McNeill 1998] McNeill, T., Gero, J. S. and Warren, J “Understanding conceptual electronic design using protocol analysis”, Research in Engineering Design, V. 10, 1998, pp 129-140.

[Meadows 1992] Meadows, D. H., Meadows, D. L., Meadows, J. R., “Beyond the limits : confronting global collapse, envisioning a sustainable future”, Chelsea Green Pub., 1992.

[Mebratu 1998] Mebratu, D, "Sustainability and sustainable development: Historical and conceptual review", Environmental Impact Assessment Review, V. 18, No. 6, 1998, pp 493-520.

[Mistree 1990] Mistree, F., Smith, W.F., Bras, B.A., Allen, J.K., Muster, D. “Decision-based design, A contemporary paradigm for ship design”, Transactions - Society of Naval Architects and Marine Engineers, V. 98, 1990, pp 565-597.

[NGM 2001] The National Geographic Magazine, November 2001.

[Oksana 2000] Oksana, M., "Product Service Systems – Final Report", The International Institute of Industrial Environmental Economics, Lund University, Sweden, 2000.

[Olewnik 2003] Olewnik, A. T, Lewis, K. E., “On validating design decision methodologies”, Proceedings of the ASME Design Engineering Technical Conference, V. 3, 2003, pp 747-756.

[Osborn 1957] Osborn A. F., “Applied Imagination, principles and procedures of creative thinking”, Scribner, NY 1957.

[Ostrosi 2003] Ostrosi, E., Ferney, M., Garro, O., "A fractal-based approach for concurrent engineering", Concurrent Engineering Research and Applications, V. 11, No. 4, 2003, pp 249-265.

[Otto 1994] Otto, K., "Robust product design using manufacturing adjustments", American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, 6th International Conference on Design Theory and Methodology, V. 68, 1994, pp1-8.

[Pahl 1988] Pahl, G., Beitz, W., "Engineering design : a systematic approach", translated by Arnold Pomerans and Ken Wallace , edited by Ken Wallace, London, Springer-Verlag, 1988.

[Pender 2001] Pender, S., "Managing incomplete knowledge: Why risk management is not sufficient", International Journal of Project Management, V. 19, No. 2, 2001, pp 79-87.

[Pine 1993] Pine B.J., "Mass Customization, The New Frontier in Business Competition", Harvard Business School Press,Boston, MA, 1993.

[Pratt 1993] Pratt, M. K., Deutsch, P. B., “Application of concurrent engineering to hi-tech product realization”, American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 65, part 1: Advances in Design Automation, 1993, pp 43-50.

[Rai 2003] Rai, R., Allada, V., “Modular product family design: Agent-based pareto-optimization and quality loss function-based post-optimal analysis”, International Journal of Production Research, V. 41, No. 17, 2003, pp 4075-4098.

[Ramaswamy 1992] Ramaswamy, R., Ulrich, K., "Augmenting the House of Quality with engineering models", American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 42, 1992, pp 309-316.

[Reich 1992] Reich, Y., 1992, "Transcending the Theory-Practice Problem of Technology," Technical Report EDRC 12-51-92, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.

[Reich 1995]Reich Y., “A critical review of General Design Theory”, Research in Engineering Design, 7(1): PP 1-18, 1995.

[Rifera 1999] Rivera-Becerra, A., Lin, L. “Measuring environmental consciousness in product design and manufacturing”, Concurrent Engineering Research and Applications,

V. 7, No. 2, 1999, pp 123-138.

[Rigo 2003] Rigo, Ph., Karr, D.G., Na, S.S., Sarghiuta, R., "Comparison of rational-based and rule-based optimum design of ship structures", Transactions - Society of Naval Architects and Marine Engineers, V. 110, 2003, pp 435-451.

[Sanchez 2002] Sanchez, R., "Fitting together a modular approach", Manufacturing Engineer, V. 81, No. 5, 2002, pp 216-218.

[Sand 2002] Sand, J., Gu, P., Watson, G., "HOME: House of modular enhancement - A tool for modular product redesign", Concurrent Engineering Research and Applications, V. 10, No. 2, June 2002, pp 153-164.

[Seliger 1997] Seliger G., Müller K., Perlewitz H., "More Use with fewer Resources", Proceedings of the 4th CIRP International Seminar on Life Cycle Engineering 1997, Berlin, pp 3-16.

[Seliger 1998] Seliger, G., Grudzien, W., Müller, K. "The acquiring and handling of devaluation." Proceedings of the 5th CIRP International Seminar on Life Cycle Engineering 1998, Stockholm (Sweden), pp 99-108.

[Seliger 2000] Seliger, G., Basdere, B., S. Schumann, and M. Wapler: "Technological Conditions for Selling Use instead of Selling Products," in Kimura, F. (ed.), the Proceeding of the 7th CIRP International Seminar on Life Cycle Engineering, November 27-29, 2000, The University of Tokyo, Tokyo, 2000, pp. 17-22.

[Simon 1969] Simon, Herbert, "The Sciences of the Artificial", Oxford University Press, 1969.

[Simpson 1998] Simpson, T.W., Rosen, D., Allen, J.K., Mistree, F., "Metrics for assessing design freedom and information certainty in the early stages of design", Journal of Mechanical Design, Transactions of the ASME, V. 120, No. 4, 1998, pp 628-635.

[Simpson 2001] Simpson, T.W., Maier, J.R.A., Mistree, F., "Product platform design: Method and application", Research in Engineering Design - Theory, Applications, and Concurrent Engineering, V.13, No. 1, 2001, pp 2-22.

[Simpson 2003] Simpson, T. W., "Product platform design and optimization: Status and promise", Proceedings of the ASME Design Engineering Technical Conference, V. 2A, 2003, pp 131-142.

[Sosale 1997] Sosale, S., Hashemian, M., Gu, P., "Product modularization for reuse and recycling", American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 94, Concurrent Product Design and Environmentally Conscious Manufacturing, 1997, pp 195-206.

[Stanley 2001] Stanley, R., "Enumerative Combinatorics", Cambridge University Press, 2001.

[Strawbridge 2002] Strawbridge, Zeke, McAdams, Daniel A., Stone, Robert B., "A computational approach to conceptual design", Proceedings of the ASME Design Engineering Technical Conference, V. 4, 2002, pp 15-25.

[Suh 1990] Suh, N. P., "The principles of design", New York, Oxford University Press, 1990.

[Suh 2001] Suh, N. P., "Axiomatic Design", New York, Oxford University Press, 2001.

[Sydenham 2004] Sydenham, P. H., "Systems Approach to Engineering Design", Artech House Inc., Boston, London, 2004.

[Taguchi 1987] Taguchi, G., "System of Experimental Design", vol. I and vol. 2, UN1PUB-Kraus International Publications (White Plains) and American Supplier Institute, Inc. (Dearborn), 1987.

[Tate 1995] Tate D., Nordlund M., "Synergies Between American and European Approaches to Design", Proceedings of the First World Conference on Integrated Design and Process Technology (IDPT-Vol. 1), Society for Design and Process Science, Austin, TX, pp. 103-111, December 7-9, 1995.

[Tebay 1984] Tebay, R., Atherton, J., and Wearne, S.H., "Mechanical Engineering Design Decisions: Instances of Practice Compared with Theory," Proceedings of the Institute for Mechanical Engineers, Vol. 1988, No. 6, 1984.

[Tipnis 1998] Tipnis, V. A., "Evolving Issues in Product Life Cycle Design: Design for Sustainability", in Handbook of Life Cycle Engineering, Kluwer Academic Publications, 1998, pp 413-460.

[Tolle 2003] Tolle, M., Vesterager, J., "VEM: Virtual enterprise methodology", VTT Symposium (Valtion Teknillinen Tutkimuskeskus), No. 224, 2003, pp 53-70.

[Tomiyama 1993] Tomiyama, T., Umeda, Y., Yoshikawa, H., "CAD for functional design", CIRP Annals, V. 42, No. 1, 1993, pp 143-146.

[Tomiyama 1994] Tomiyama, T., "From general design theory to knowledge-intensive engineering", Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, V. 8, No. 4, 1994, pp 319-333.

[Tomiyama 1995] T. Tomiyama, T. Sakao, Y. Umeda, Y. Baba, "The Post-Mass

Production Paradigm, Knowledge Intensive Engineering, and Soft Machines,”, in Krause, F.-L. and H. Jansen (eds.), Life Cycle Modelling for Innovative Products and Processes, Chapman & Hall, London, 1995, pp. 369-380.

[Tomiyama 1996] Tomiyama T, Yoshikawa H., "Extended general design theory, Design Theory for CAD", Elsevier Science Publishers, 1987. pp 95-130.

[Tomiyama 1997] Tomiyama, T., “A Manufacturing Paradigm Toward the 21st Century,” Integrated Computer Aided Engineering, Vol. 4, 1997, pp. 159-178.

[Tomiyama 1999] Tomiyama, T., “The Post Mass Production Paradigm,” in Yoshikawa, H., R. Yamamoto, F. Kimura, T. Suga, and Y. Umeda (eds.), Proceedings of the First International Symposium on Environmentally Conscious Design and Inverse Manufacturing (EcoDesign ’99), February 1-3, 1999, Tokyo, IEEE Computer Society, 1999, pp. 162-167.

[Tomiyama 2000] Tomiyama, T., “Knowledge Intensive Engineering towards Sustainable Products with High Knowledge and Service Contents,” in Horváth, I., A.J. Medland, and J.S.M. Vergeest (eds.), TMCE 2000, Third International Symposium on Tools and Methods of Competitive Engineering, April 18-20, 2000, Delft University Press, Delft, the Netherlands, 2000, pp. 55-67.

[Tribus 1969] Tribus, M., 1969, “Rational Descriptions, Decisions and Designs”, Pergamon Press, Elmsford, NY.

[Tsai 1999] Tsai, Y., Wang, K., “Development of modular-based design in considering technology complexity”, European Journal of Operational Research, V. 119, No 3, 1999, pp 692-703.

[Tsai 2004] Tsai, H., Hsiao, S., "Evaluation of alternatives for product customization using fuzzy logic", Information Sciences, V. 158, No. 1-4, 2004, pp 233-262.

[Tseng 1996] Tseng, M. M., Jiao, J., Merchant, M. E., "Design for mass customization", CIRP Annals - Manufacturing Technology, V. 45, No. 1, 1996, pp 153-156.

[Tseng 1998] Tseng, M., Du, X., "Design by customers for mass customization products", CIRP Annals - Manufacturing Technology, V 47, No. 1, 1998, pp 103-106.

[Ullman 1987] Ullman, D.G., Stauffer, L.A., and Dietterich, T.G., "Preliminary Results of an Experimental Study of the Mechanical Design Process," Proceedings from the NSF Workshop on the Design Process, Waldron, M.B., ed., Ohio State University, Oakland, CA, February 8-10, 1987, pp. 145- 188.

[Ullman 1992] Ullman, D. G. ,"The Mechanical Design Process", McGraw-Hill Inc., 1992.

[Ullman 1994] Ullman, David G., "Issues critical to the development of design history, design rationale and design intent systems", American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V.68, 1994, pp 249-258.

[Ulrich 1991] Ulrich, K., Tung, K., "Fundamentals of product modularity", American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, V. 39, Issues in Design/Manufacture Integration, 1991, pp 73-79.

[Vancouver 2000] City of Vancouver, "Adaptable Design Guidelines", http://www.city.north-van.bc.ca/Housing.

[Wallace 2000] Wallace, K.M., Blessing, L.T.M., "Observations on some German contributions to engineering design in memory of professor Wolfgang Beitz", Research in Engineering Design - Theory, Applications, and Concurrent Engineering, V. 12, No. 1, 2000, pp 2-7.

[Wassenaar 2001] Wassenaar, Henk Jan, Chen, Wei, “An approach to decision-based design”, Proceedings of the ASME Design Engineering Technical Conference, V. 4, 2001, pp 21-30.

[WBCSD 1998] WBCSD, “Cleaner Production and Eco-Efficiency – Complementary Approaches to Sustainable Development”, The World Business Council for Sustainable Development and United Nations Environment Program – Industry and Environment, Paris, 1998.

[Webb 2002] Webb, Alan, "TRIZ: An inventive approach to invention", Manufacturing Engineer, V. 81, No. 4, 2002, pp 171-177.

[Whitney 1993] Whitney D., Nevins J., DeFazio T., Gustavson R., "Problems and Issues in Design and Manufacture of Complex Electromechanical Systems", C. S. Draper Lab Report R-2577, December 1993.

[Willems 2003] Willems, B., Seliger, G., Duflou, J., Basdere, B., “Contribution to Design for Adaptation: Method to Assess the Adaptability of Products (MAAP)”, Proceedings of EcoDesign2003: Third International Symposium on Environmentally Conscious Design and Inverse Manufacturing, Tokyo, Japan, December 8-11,2003.

[Yoshikawa 1981] Yoshikawa, H., "General Design Theory and a CAD System", Man-Machine Communication in CADICAM. Proceedings of the IFIP WGS.2/5.3 Working Conference 1980, Tokyo, North Holland, Amsterdam, 1981, pp. 35-58.

[Yoshimura 2003] Yoshimura, M., Izui, K., Fujimi, Y., "Optimizing the decision-making process for large-scale design problems according to criteria interrelationships", International Journal of Production Research, V.41, No. 9, 2003, pp 1987-2002.

[Yu 1999] Yu, J.S., Gonzalez-Zugasti, J.P., Otto, K.N., "Product architecture definition based upon customer demands", Journal of Mechanical Design, Transactions of the ASME, V. 121, No. 3, 1999, pp 329-335.

[Yu 2003] Yu, T., Yassine, A., Goldberg, D. E., "A genetic algorithm for developing modular product architectures", Proceedings of the ASME Design Engineering Technical Conference, V. 3, 2003, pp 515-524.

[Zeng 1999-a] Zeng, Y., Gu, P., "Science-based approach to product design theory. Part I: Formulation and formalization of design process", Robotics and Computer-Integrated Manufacturing, V. 15, No. 4, 1999, pp 331-339.

[Zeng 1999-b] Zeng Y, Gu P., "A science-based approach to product design theory. Part II: formulation of design requirements and products", Robotics and Computer Integrated Manufacturing, V. 15, No. 4, 1999, pp 314-352.

[Zha 2002] Zha, Xuan F., Lu, Wen F., "Knowledge intensive support for product family design", Proceedings of the ASME Design Engineering Technical Conference, 28th Design Automation Conference, V. 2, 2002, pp 603-612.

[Zwicky 1948] Zwicky, F., “Morphological Analysis and Construction”, Wiley Inter-science, New York, 1948.

[Zwicky 1969] Zwicky, F., “Discovery, Invention, Research through Morphological Analysis”, MacMillan, New York, 1969.

Appendix 1: A Function Representation Scheme for Conceptual Mechanical Design

The *function* of a designed object represents the intent or purpose of creating that object in terms of what it must do. The intent of the design object depends on the design domain. For example, in the design of an organization the purpose might be to provide a certain type of service to customers. In the context of mechanical design, which without mentioning will be the context of all discussions hereafter, the term "function" refers to a *physical function*, as defined below:

Definition: The function of a device in mechanical engineering design is to effect its physical surrounding in an intended manner.

From this definition we will draw a guideline to determine whether a requirement is or is not function in the context of mechanical engineering design.

Guideline: A non-physical requirement such as "cost", in the context of mechanical engineering, is not a function. A requirement is a physical requirement (a function) if it can be described as some effects on the physical environment that consists of material, energy, signal, and force.

Thus, in order to define the function of an object we must describe both the surrounding of the object (*physical environment*) and the *effects* of the object on the physical environment. This appendix presents a taxonomy of *entities* to describe the physical world, and a taxonomy of different types of effects that an object may have on its physical environment. Then it shows a scheme for the construction and representation of mechanical functions using entities chosen from these taxonomies.

A1.1. Function Operands (Physical Entities)

A function is defined as the effect of an object on some entities that describe the physical environment around the object. The entities whose attributes are affected by an object are called "*function operands*". The initial and final states of function operands are called the inputs and outputs of a function; inputs and outputs are not necessarily physical entities that enter or exit the design object. Therefore, the definition of a function requires the identification of the entities that sufficiently describe the physical world for the purpose of mechanical design.

First, "*material*" is recognized as a category of physical entities. This category presents all physical objects in the environment, including solids, liquids, and gases. The second type of entity is "*energy*". Energy in some cases can be treated as an attribute of material. For instance, the thermal energy of a hot piece of metal may be considered an attribute of that material object. However, energy might not require a material carrier, thus cannot be expressed as an attribute of material (e.g. the electro-magnetic energy). Also, in many cases the material carrier is of no importance to the design task. For example, the input to a gearbox is better described as rotational energy than described as a shaft with rotational energy as one of its attributes. For these reasons energy is considered an independent category of physical entities.

The third category recognized is “*force*”, which is a measurable phenomenon observed as pulling or pushing things. Force cannot be described by materials, thus this category is independent from the first category. However, it is often considered as belonging to the category of energy in the design literature [Pahl 1988]. A reason for not considering force as an independent category is that the practical application of “force” in engineering is typically associated with ‘displacement’, thus force can be described by energy. Similar reasoning may also question the independence between materials and energies. These discussions are avoided here because the classification of entities may be made arbitrarily for the purpose of developing a practical representation scheme for mechanical functions.

The above three categories can describe a broad range of entities. However, physical objects usually are not disconnected entities without any "order". The arrangement of multiple objects, their shapes, and the temporal sequence of events carry "*information*". In engineering design, the information conveyed in the existing order among and within entities represents the human notion of the meaning, behavior, or purpose of entities. For example, a word written on a piece of paper contains information; which is a particular order between the material entities (the paper and ink). Information usually requires a physical carrier in the form of material or energy; however the carrier itself may not be important to the design problem. Therefore, information is recognized as an independent category, which is not expressed by the previous three categories. In Systematic Design, information together with its physical carrier is called "*signal*" [Pahl 1988]. By this definition, signal is the physical manifestation of information; signal refers to a certain order that exists in the properties and attributes of entities and events.

It is necessary to discuss whether or not "time" should be treated as an independent entity. There is no device whose function is to affect time; therefore time cannot be a function operand and is not considered an independent entity. Instead, time is an important part of "*actions*" that describe the effect of an object on its physical environment. Actions and the role of time in describing them will be discussed in the next section.

In the rest of this appendix, we refer to the above four basic categories, *material*, *energy*, *force* and *signal*, as "*principal entities*" or *PEs* for short. PEs in turn can be sub-divided into more specific types. For example material can be solid or liquid or gas. PEs and their sub-types form a taxonomy of function operands. Figure A1.1 shows a prototype taxonomy of function operands developed for the description of mechanical functions.

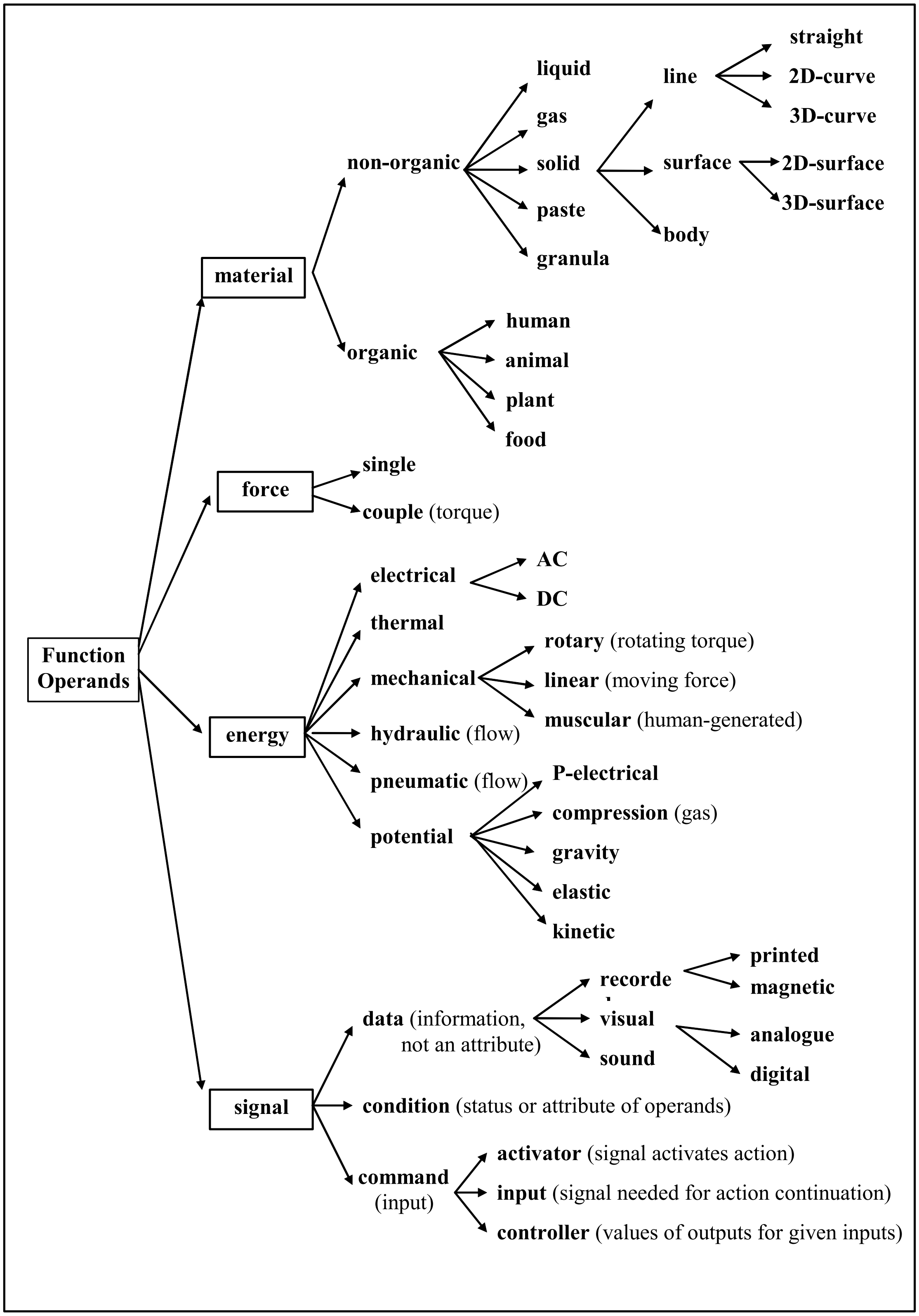


Figure A1. 1: The hierarchical taxonomy of function operands.

Locating an entity in the above taxonomy does not sufficiently describe it. To describe entities precisely, either the sub-division of types must continue into minute details, or the characteristics of entities must be further specified by their *attributes*. Exhaustive sub-division is not practical since for almost every new entity a new sub-category must be defined which requires the continuous modification of the taxonomy. On the other hand, using attributes is advantageous since attributes (e.g. velocity, length, weight, dimension, color, smell, frequency, taste, magnitude, direction, etc.) can be shared by all relevant entities.

We have developed a pool of attributes, where each attribute is represented by a list of "item-value" pairs. First an entity (function operand) is selected from the above taxonomy. Then it is further specified via selecting its relevant attributes from the attribute pool and evaluating the values for attributes. For example the entities "rotational mechanical energy" and "metal bar" require attributes "torque" and "length" respectively.

Table A1.1 shows some of the attributes developed for entities. The column "type" shows whether an attribute is specified by numerical or symbolic values. The column "representation" indicates how the value of an attribute is represented in our scheme, e.g. by lower and upper limits or by *strings*. The column "access" shows the rules of accessing operands in databases. For example, numerical attributes can be accessed by "greater than" or by their exact values. The column "taxonomy" shows where in the operand taxonomy (and all branches thereafter) the operand is applicable. In the "Type" column of the table, *n[L-U]* indicates that the attribute value is numerical and is presented by lower and upper limits; *strings* are chosen from a set of pre-specified words; *code string* is a particular type of string which is either generated by the standard vocabulary or is copied from an *operand* within the function.

We have found it more practical not to consider the "*location*" of an entity as one of its attributes; otherwise we would have to define a fixed spatial reference in order to specify the location of entities. The absolute location of entities with respect to a fixed reference is rarely of importance in mechanical design problems. Instead, a change in location (or prevention of change) is usually important. We deal with location in the "action" part of a function, to be discussed in the next section.

The four categories of entities in Figure A1.1, together with their attributes in Table A1.1, can generally describe most entities of interest in conceptual mechanical design. Since the location of an entity is not considered as one of its attributes, an entity may be described by its type in the PE taxonomy, the value of its relevant attributes, and its location.

Attribute	Type	Access	Taxonomy	Comments
mass	n, [L - U]	> < =	material, gravity	
corrosion	string	Boolean	liquid, gas	
shape	string	Boolean	Solid-body	
contour	string	Boolean	2D-surface	circle, square, etc.
density	n, [L - U]	> < =	liquid, gas	
viscosity	n, [L - U]	> < =	liquid	
toxicity	String	Boolean	material-non-organic	
length	n, [L - U]	> < =	line	
diameter	n, [L - U]	> < =	line	
volume	n, [L - U]	> < =	material	
area	n, [L - U]	> < =	body, surface	
grain-size	String	Boolean	granular	
tangling	String	Boolean	granular	
elasticity	n, [L - U]	> < =	solid	
strength	n, [L - U]	> < =	solid	
material-type	String	Boolean	material-non-organic	wood, plastic, metal, …
load-magnitude	n, [L - U]	> < =	force-single, mech-linear, muscular	
nature			force	type of force
torque-magnitude	n, [L - U]	> < =	force-couple, mech-rotary	
direction	String	Boolean	force	
oscillation	n, [L - U]	> < =	force	frequency of direction variation
velocity	n, [L - U]	> < =	energy-mech.	
elevation	n, [L - U]	> < =	gravity	
deflection	n, [L - U]	> < =	elastic	
magnitude	n, [L - U]	> < =	energy	
power	n, [L - U]	> < =	energy	energy-rate
voltage	n, [L - U]	> < =	electrical	
current	n, [L - U]	> < =	electrical	
temperature	n, [L - U]	> < =	material, thermal	
pressure	n, [L - U]	> < =	mat-liquid, mat-gas, hydraulic, pneumatic, compression	
flow rate	n, [L - U]	> < =	pneumatic, hydraulic	
target-attribute	code string	=	signal-condition	specifies the target condition that is being monitored
medium	code string	=	signal-command	carrier of a signal

Table A1. 1: The list of operand attributes.

A1.2. Actions

We regard the function of an object as its effect on its physical surrounding. This effect might be to change the characteristics of *function operands* or it might be to prevent them from changing. In our scheme, the part of a function that describes such effects is called the "*action*". An action affects the properties of one or multiple function operands, which are physical entities. It was mentioned in the previous section that entities are specified by their *types*, *attributes*; and *location*. Since an action may affect any of these parameters, three types of actions can be identified: *change-type*, *change-attributes*, and *change-locations* of entities.

Also, for two reasons we introduce the action "*change-number*" (connect or separate) as a distinct type of action. First, entities from the categories of "energy" and "signal" may need physical carriers. It may be necessary to add the carriers to these entities in the functional structure; or it may be necessary to remove the carriers. The addition or removal of entities can be achieved via the actions “connect” and “separate”. For example, when mechanical energy contracts a spring, we may consider this function as "connect energy and material". Second, the function of a device can be to split or join entities in circumstances that keeping track of the original entities is no longer useful to the design task. For example, consider a machine that mixes two substances for further processing in a larger system. The mixture may be then represented as one entity in the function structure of the overall system when there is no need to trace its ingredients. The action “connect” represents the function of the mixer, which is to replace the initial state (two entities) with the final state (one entity) in the functional structure.

Therefore, this function representation scheme recognizes four main types of actions:

change_type, *change_attribute*, *change_location*, and *change_number*. The opposite of an action is also considered an action. That is, '*prevent from change'* may be applied to the type, attribute, location, and number of entities. For example, the action of a thermal insulation material can be described as "prevent from change in location". Further detailing of *action types* generates a hierarchical taxonomy similar to the one developed for operands. The taxonomy of actions is shown in Figure A1.2.

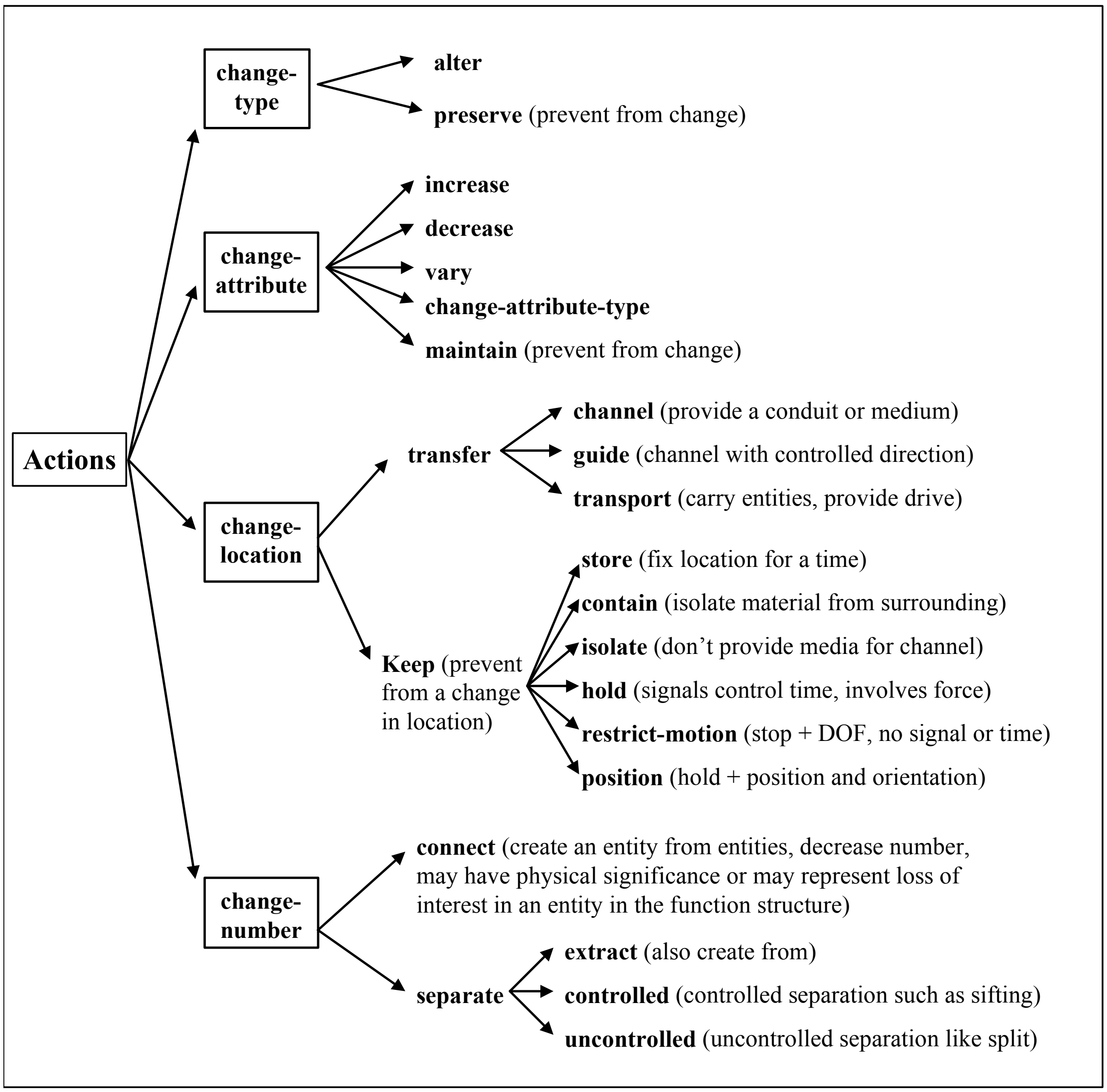


Figure A1.2: The taxonomy of actions.

In the same way principal entities were further specified by their attributes, actions are further specified by their "*specifiers*". Action specifiers explain the way an action is performed. For instance, "*distance*: 2 meters", "*time*: 12 seconds", and "*accuracy*: 1%" are three specifiers for actions "transfer", "store", and "change-attribute" respectively. Table A1.2 shows a partial list of the specifiers developed for our pilot computer system.

Specifier	Type	Access	Taxonomy	Comments
ratio	n, [L - U]	> < =	change-attribute	
no-of-ratios	n, [L - U]	> < =	change-attribute	infinity, for continuous variation
distance	n, [L - U]	> < =	transfer	
direction	string	Boolean	change-location	X, Y, Z, CW, CCW
time	n, [L - U]	> < =	store	duration of an action
accuracy	n, [L - U]	> < =	change-attribute	values in %
DOF	string	Boolean	restrict-motion	X, Y, Z, CW, CCW
target	code string	=	change-type and change-attribute	shows what sub-type or attribute of the input changes
activating-signal	code string	=		
utilized-energy	code string	=		

Table A1. 2: The list of action specifiers.

Figure A1.3 summarizes the elements of function operands and actions. A function operand is described by its *type*, *attributes* and its "*connection code*". Connection code describes how the operand is connected to its action, and explains the role of an operand of a function within a functional structure. For example, if the connection code of an operand is “input”, this operand represents the initial state of the physical environment; if the code is “output”, the operand represents the final state of the physical environment after the function has been executed. Thus the relation between actions and operands is described as a part of ‘operands’, and an action can be simply described by its two elements: its *type* chosen from the action taxonomy, and its *specifiers* chosen from the list of available specifiers.

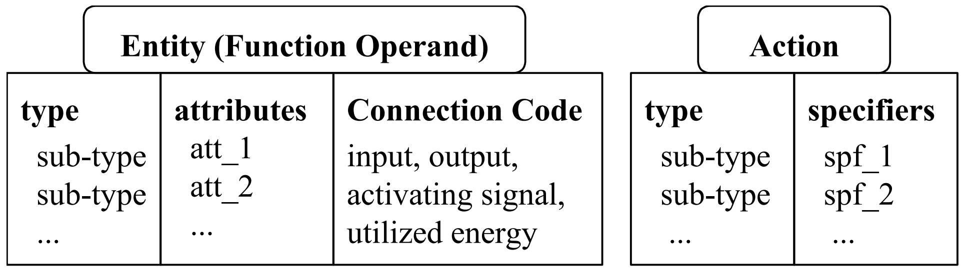


Figure A1.3: The constituting elements of function operands and actions.

A1.3. The Structure of a Function

The previous two sections described two constituting elements of functions: *function operands* and *actions*. Function operands represent the physical entities that are affected by the function of an artifact; an action describes the effect. This section explains how these elements are assembled together to build functions.

Basic Functions

In the proposed function representation scheme, a function is constructed from certain building blocks called *basic functions*. A basic function consists of <u>one</u> action and a number of function operands. Any number of basic functions may be used to sufficiently describe a function. A simple function that can be sufficiently described by one action consists of one basic function. A simple function has all the properties of complex functions and can be decomposed or altered. Figure A1.4 shows how a basic function is constructed from actions and operands, specified by their types/specifiers and types/attributes respectively.

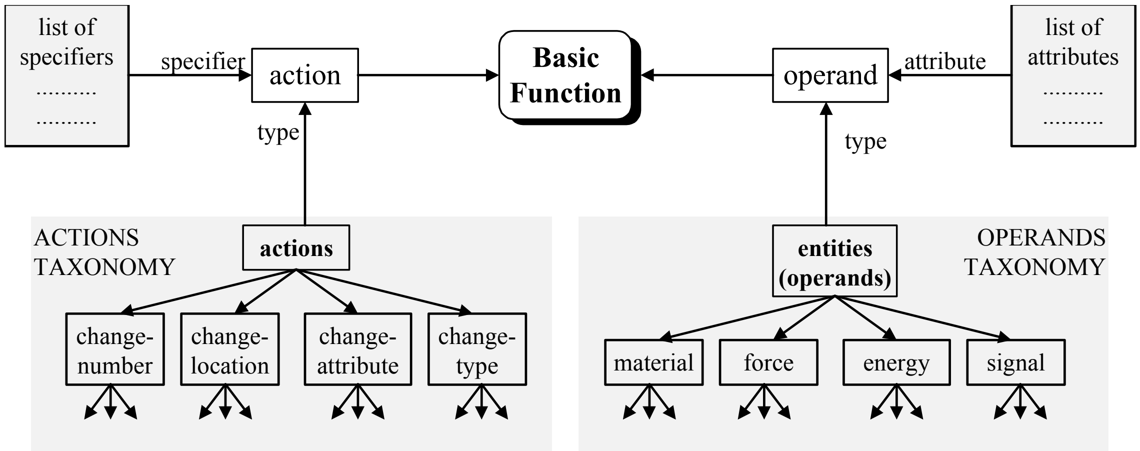
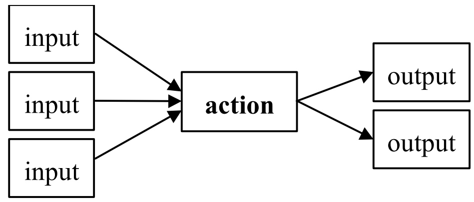


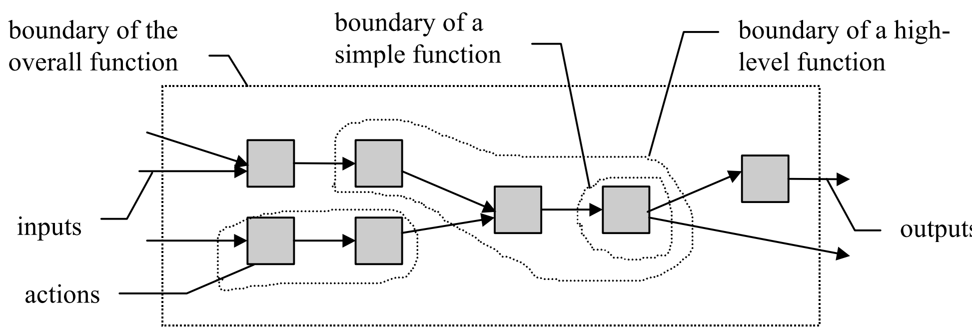
Figure A1.4: Elements of a basic function in the proposed scheme.

Constructing Larger Functions

A function may be expressed in terms of a number of basic functions and is recognized in the functional structure by its *function boundary*. Larger functions are composed of other functions. Thus, we can build a hierarchical functional structure where the *overall function* is constructed from other functions, and these functions from smaller functions down to *basic functions* at the lowest level of the hierarchy. Figure A1.5 shows the structure of a function.



a) Structure of a ***Basic Function***



b) The Overall Function of a system

Figure A1.5: Construction of functions in the proposed scheme.

In the proposed scheme a function operand (an entity) is described by its type, selected from the taxonomy, and its attributes, selected from the attribute list. There are only a small number of types and attributes to choose from. However, the designer can describe a large number of physical entities through the morphological combination of different types and attributes. Similarly, actions are constructed by their types and their specifiers. A large number of actions can be constructed through different combinations of action types and specifiers. Also, actions and operands may be combined in various ways to produce a large number of possible basic functions, and any number of basic functions can be used to describe a function. Thus, the proposed scheme provides for the possibility of morphological combination of different elements within a function. As a result, a small vocabulary can describe a large number of complex functions.

A1.4. Examples

In this section a few examples are provided to demonstrate the expressive power of the proposed function representation scheme. For each example, we consider a device and its nominal function, and then show how this function can be described using the proposed scheme.

Example 1: an analogue measurement device (e.g. a thermometer)

The inputs of this function are the temperature to be measured and a pre-specified scale. The first operand is of the type “signal-condition”. It is further specified by the attribute “target attribute”. The second input, also of type signal, is recorded data. The action is to “connect” these two signals in order to generate a new display signal. Generally, the function "*compare*" can always be described as connecting input signals to produce an output signal. Also, the function "measure" is a special case of "compare" in that a signal is compared against pre-specified units. This example is illustrated in Figure A1.6.

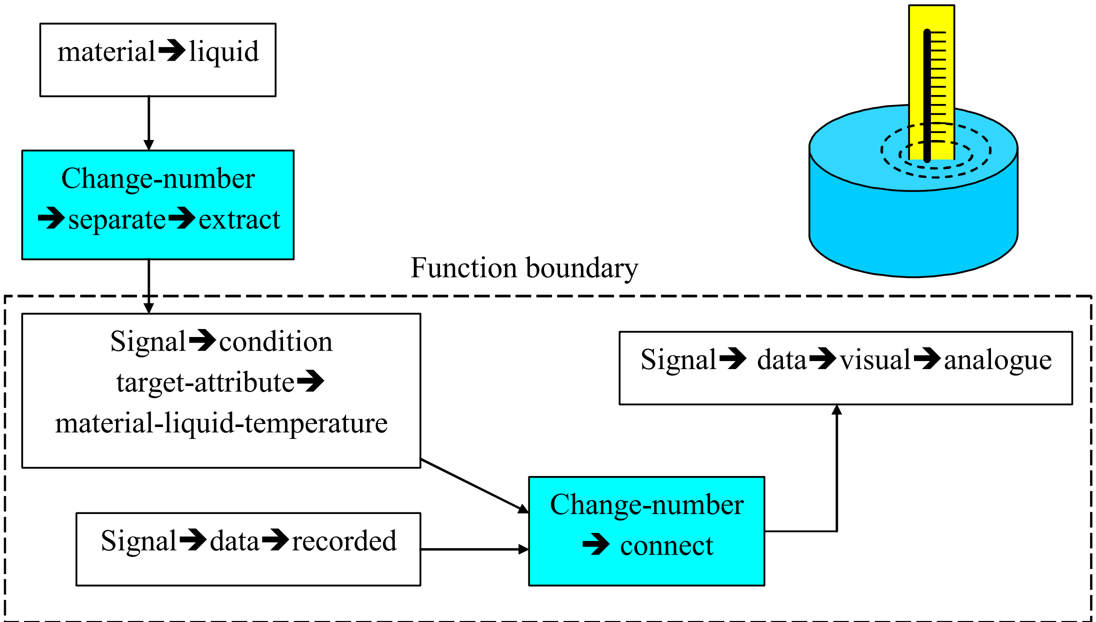


Figure A1.6: Representing the function of a thermometer.

Example 2: a current controller for a servomotor (e.g. for speed control)

This function includes the “measure” function explained above, as well as other actions to change the value of electrical energy. The function consists of three basic functions. First, the signal that is the condition of the monitored quantity (e.g. speed) is connected to a pre-specified target value to produce the difference signal. This signal is a command pulse that can have a carrier of type, for example, electrical energy. Then this signal is connected to a source of energy to release the exact amount of energy required to correct the initial value. This energy is then connected to the main source to produce the final output. This representation in general can describe the function "*control*". It is shown graphically in Figure A1.7.

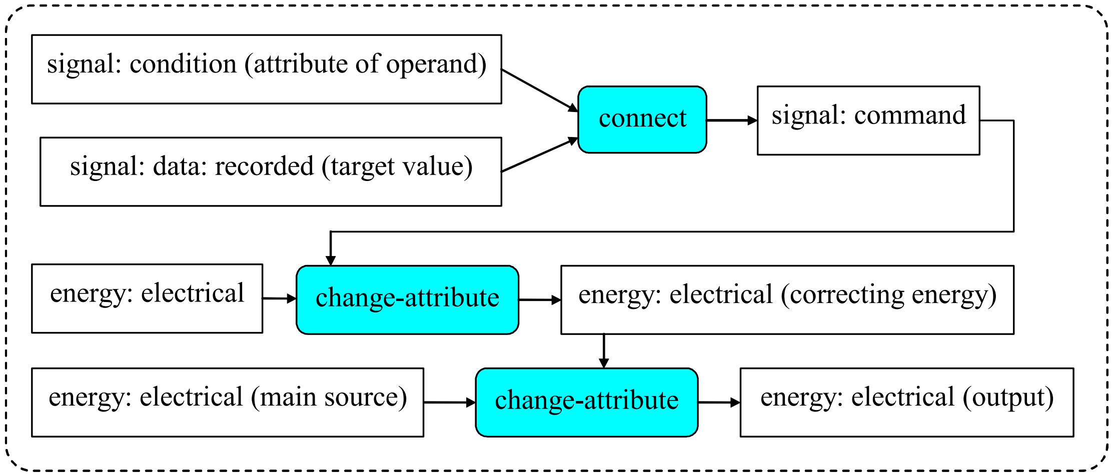


Figure A1.7: Representing the function of a controller.

Example 3: a shaft

This is a simple example that does not involve a signal, since a shaft transmits torque unconditionally. This function is represented as shown in Figure A1.8:

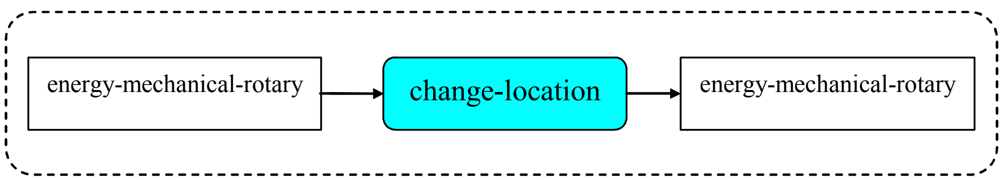


Figure A1.8: Representing the function of a shaft.

The following examples show how some common functions are represented in the proposed scheme:

- *Compare* = change-number→connect (signal + signal = signal)
- *Measure* = a type of compare
- *Control* = compare + change-attribute→vary + change-attribute→vary
- *Store* = change-location→keep→store

- *Create* = change-number→separate→extract

A1.5. Software Implementation

We have developed a pilot software system (TARRAUH) for the proposed function representation scheme. The interface includes a canvas divided into two areas as shown in the following figures. The left area of a canvas is the hierarchy of a functional structure, and the right area shows the details of functions and their constituting 'basic functions', as well as the relations between all the elements involved in the construction of functions.

A1.5.1. Operands

TARRAUH allows the user to represent an operand by its type and its attributes using the standard vocabulary. This is illustrated in Figure A1.9. Each operand is represented by a link with a red node and a blue node on its two ends. The thickness and color of a link shows the type of operand (e.g. material or energy). A link attaches to "actions" through its nodes. When the red node attaches to an action, it indicates the output of that action. When the blue node attaches to an action it indicates the input to that action. The unattached nodes are larger in size. Such large nodes indicate the main inputs and outputs of the overall function. TARRAUH pops up the proper dialogue boxes so that the user can select the right "type" of the operand from the taxonomy through menu options. This is shown in the figure.

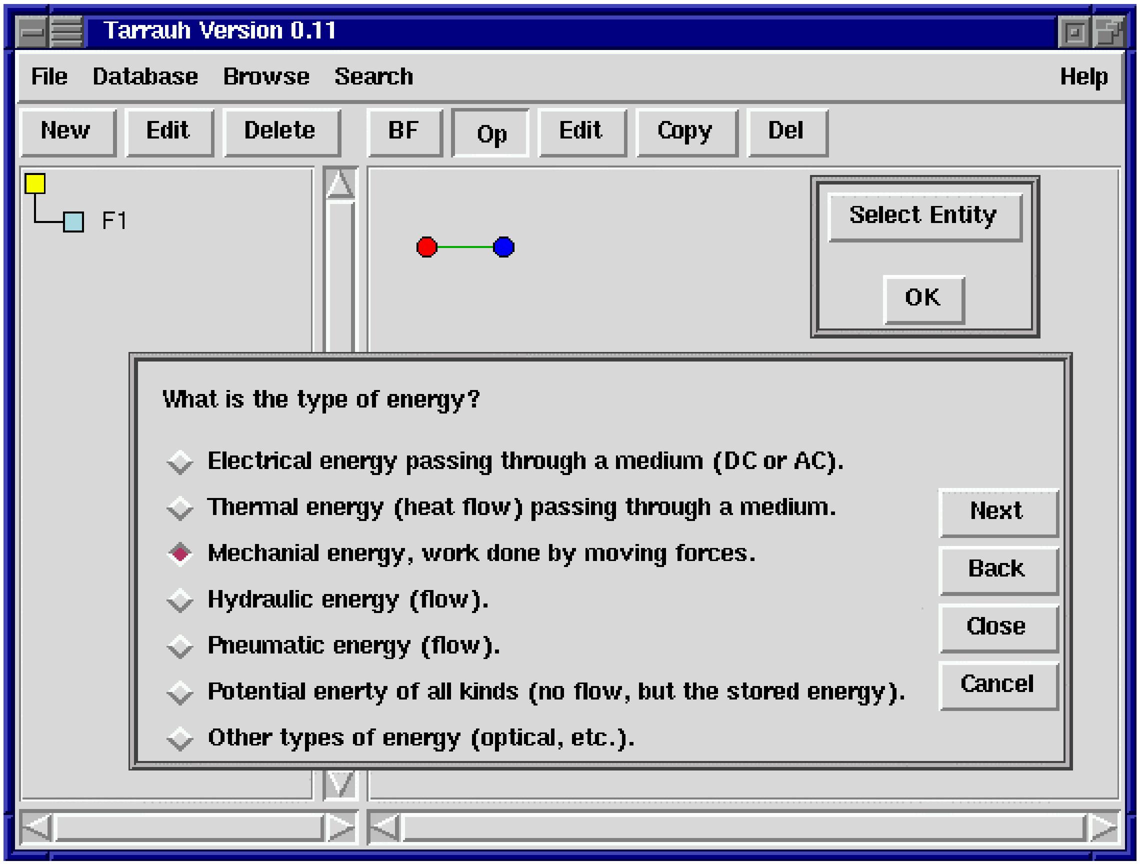


Figure A1.9: Specifying the types of function operands.

Once the type of an attribute is defined, TARRAUH automatically retrieves the relevant attributes from the attribute pool and asks the user to enter their *values*. For numerical attributes there are boxes for lower and upper limits of their values. For "string" attributes, a small pull-down menu allows the user to select a word from the list of possible strings that are available in the program. This is illustrated in Figure A1.10.

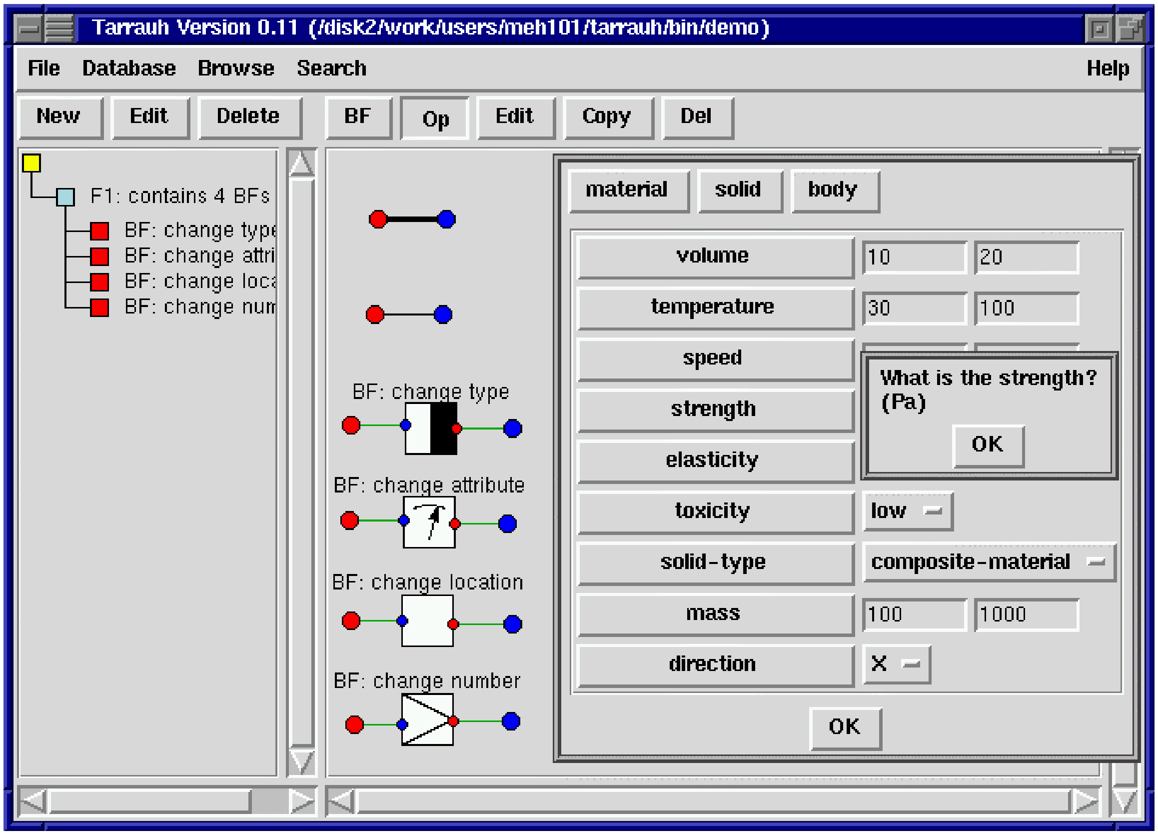


Figure A1.10: Specifying the relevant attributes of a function operand (solid, material).

A1.5.2. Actions

In TARRAUH actions are denoted by small boxes. There are four different shapes of boxes for the representation of four main actions (e.g. change-type and change-location). This is demonstrated in Figure A1.10.

TARRAUH allows the user to specify the 'type' of an action by picking the right item from within the dialogue boxes (Figure A1.11). Actions together with their operands are the building blocks for constructing functions. Thus, actions are put together to build a function. TARRAUH shows the grouping of actions under a function in the left area of the canvas as shown in Figures A1.10 and A1.11.

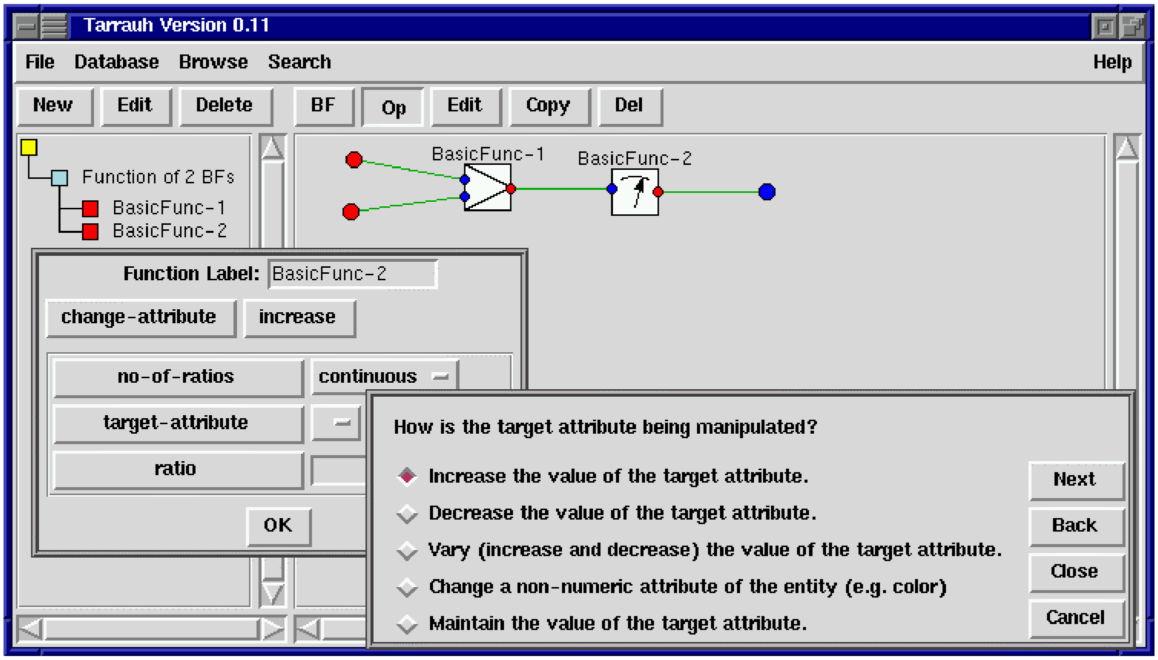


Figure A1.11: Different "types" of actions in TARRAUH.

Once the type of an action is fully specified, TARRAUH pops up the proper dialogue boxes that contain pull-down menus to select the action specifiers. This is shown in Figure A1.12. The figure shows the relevant specifiers for the action change-attribute→decrease. In this case the relevant specifiers are no-of-ratios, target-attribute, and ratio.

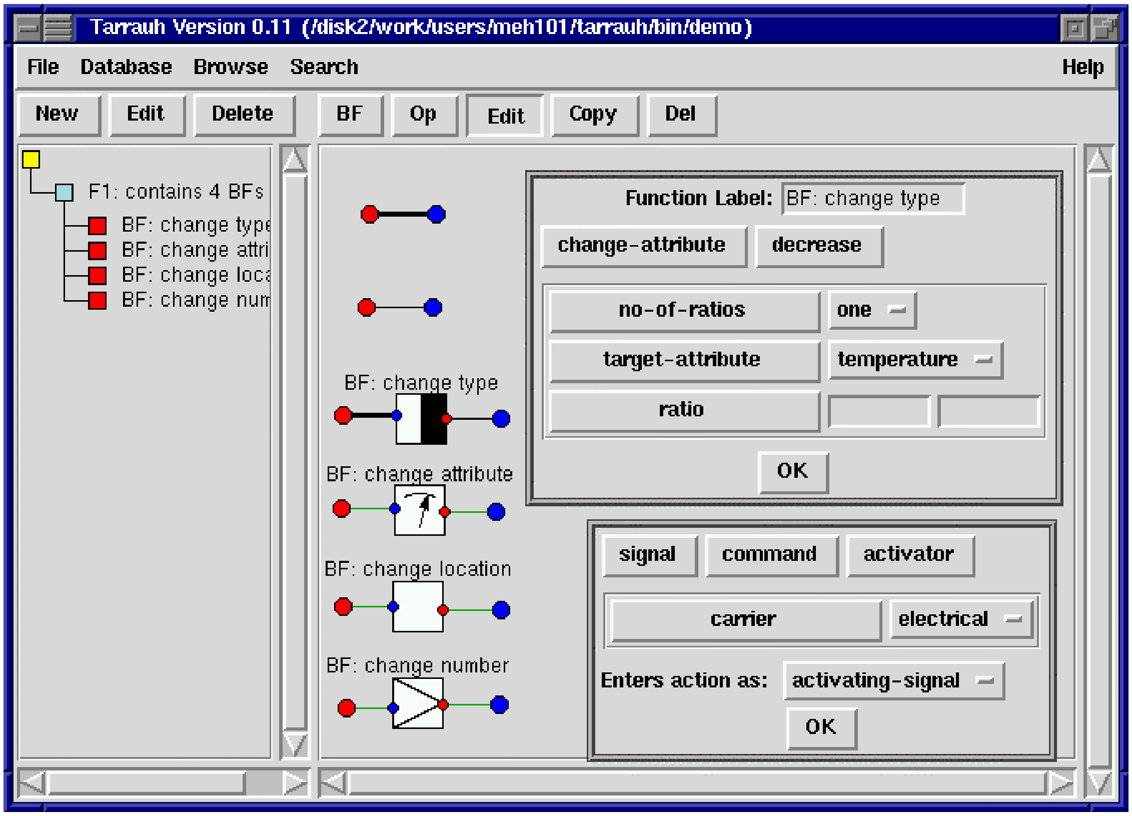


Figure A1.12: Quantifying "action specifiers" for the actions of known types.

A1.5.3. Building Larger Functions

In the proposed scheme, functions consist of building blocks and also of other smaller functions, resulting in a hierarchic functional structure. This hierarchy exists in all design problems regardless of the representation scheme that is used. Generally, in a design problem a functional structure is the result of a decomposition process. In a redesign task, however, the functional structure of a product might be initially constructed by studying an existing product. We recognize the following two distinct and equally important characteristics in a functional structure.

First, a functional structure is a *graph* that can be also called a *network*. The nodes of this network are functions at various levels of the hierarchy down to the end-node functions. The "end" level of decomposition is where functions can be systematically replaced by readily-available solutions, that is, the solutions which can be obtained from outside sources and their further decomposition is not a part of the design task in hand. The *links* within this network represent different types of functional interactions among functions or *function carriers* (solutions).

Second, a functional structure is a hierarchical structure that can be called a *tree*. This tree shows the decomposition history hence the design rationale. The initial functional requirements of a design problem are the root of the tree. These complex functions may require complex solutions which are not readily available; therefore they are decomposed into the functional requirements of their solutions. The decomposition process continues until readily-available solutions for all functions are found.

Figure A1.13 schematically shows these two important aspects of a functional structure. The network is represented as several end-node functions that are linked together through their sharing operands. The tree is represented by a series of nested function boundaries. From a top-down view, the tree is the nested segmentation of the overall functions; and from a bottom-up view, it is a recursive grouping of smaller functions.

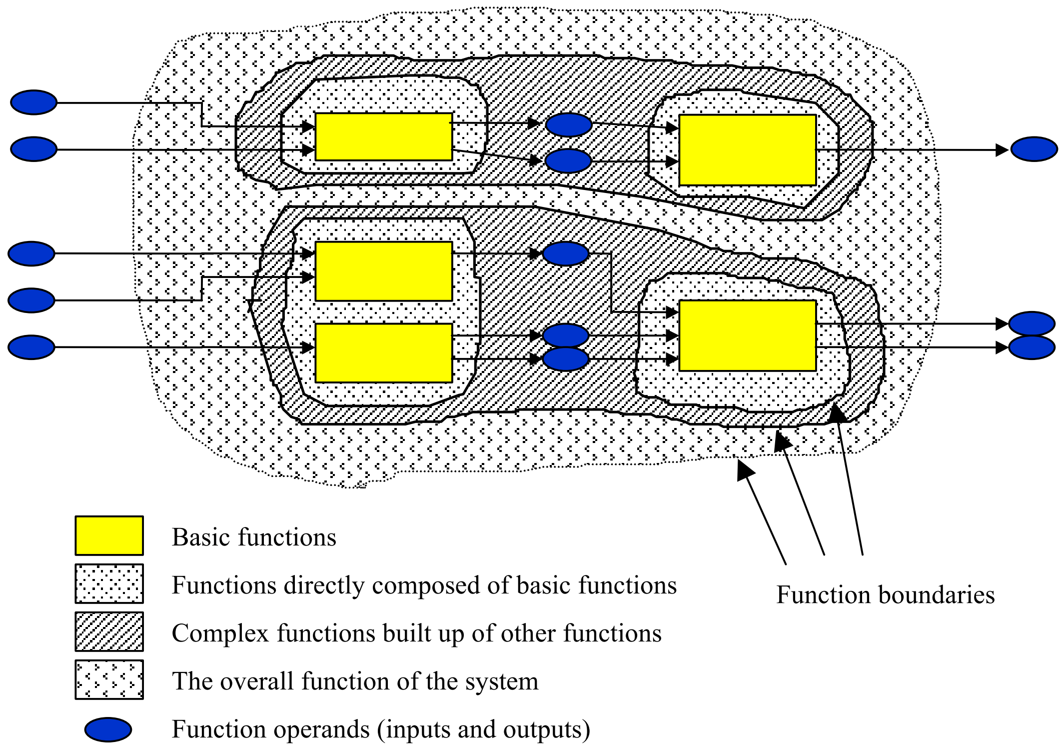


Figure A1.13: Representing networks and trees in a functional structure.

TARRAUH is capable of representing both the tree and the network in a functional structure. The right area of the canvas represents the network. The left side represents the decomposition tree. The functions on the left are directly linked with the "basic functions" on the right. Any changes in either side are automatically reflected in the other. Functions can be moved and regrouped on the left side. This is equivalent to shifting function boundaries in the above figure.

The following figure shows how the function of a car, which is to transport people and objects, is represented and decomposed to form a functional structure. The decomposition history is represented on the left. The function network is represented on the right by actions and operands.

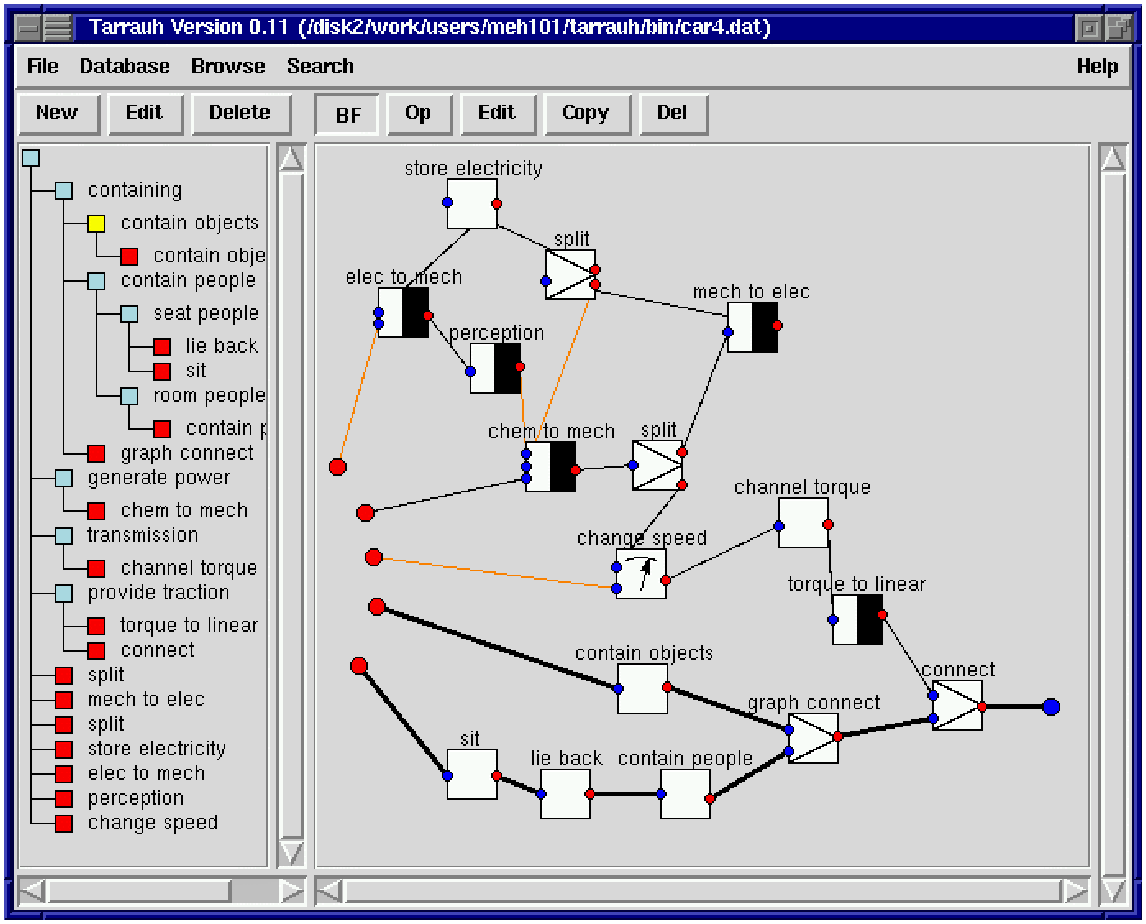


Figure A1.14: Representation of the function of a car in TARRAUH.

Appendix 2: Four-Wheel Servo Steering

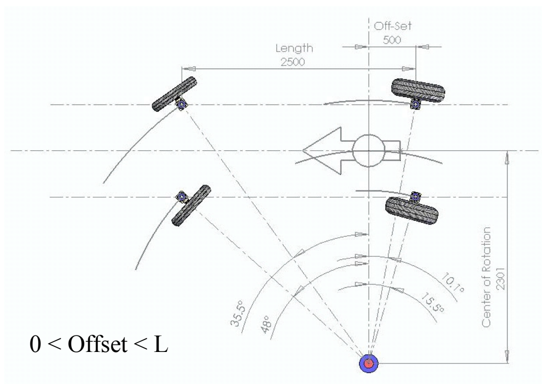
This appendix discusses some added capabilities which can be made possible in a four-wheel servo steering vehicle.

A2.1. Steering Axis Offset

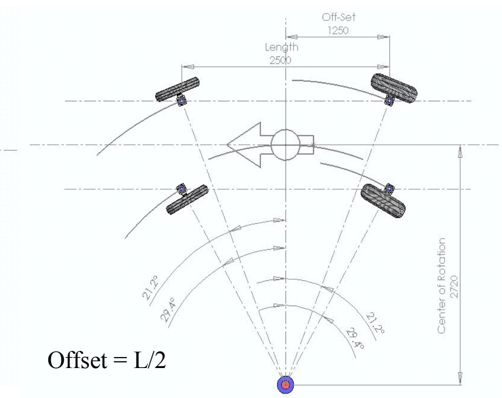
In front wheel steering, which was depicted in Figure 5.17 of Chapter 5, the *instantaneous center of rotation* (ICR) is always located on a line that connects the rear wheels. Thus, steering action can be translated to the shifting of the location of ICR on this line. The distance between ICR and the centerline of the vehicle is the only variable parameter in steering. This distance varies between infinity (vehicle moving straight) and the minimum turning radius (when the driver steers to the maximum).

When rear wheels also turn, ICR can be anywhere on the two dimensional plane. If we call the line drawn from ICR perpendicular to the vehicle's centerline *'the steering axis'*, then two variable parameters can be defined for this steering mechanism. One is the location of the steering axis on the vehicle's centerline, and the other one is the location of ICR on this axis.

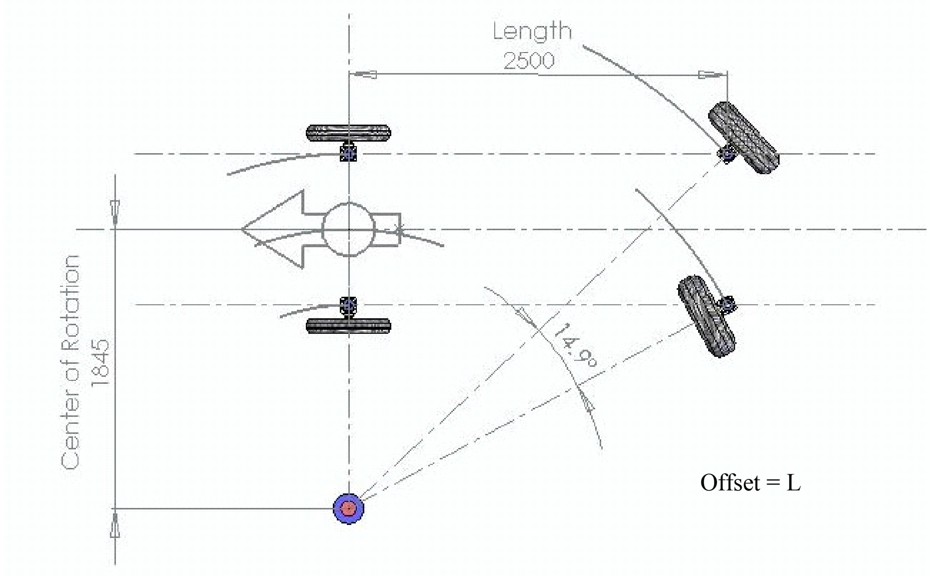
Here we measure the location of the steering axis on the vehicle's centerline from the rear axis, and call this parameter the '*offset'*. When the offset is zero, the vehicle behaves like a regular front wheel steering car. When the offset equals the distance between the front and rear axes, denoted by L, the vehicle behaves like a warehouse forklift. Figure A2.1 shows the shifting of the steering axis and the effect of this offset on steering angles.



Steering axis shifted forward



Steering axis in the middle



Rear steering

Figure A2. 15: The offsetting of the steering axis.

Since there are two variable parameters in this system, there could also be two driver control devices. One is the steering wheel which determines the location of the ICR on the steering axis, and the other is a shifter which determines the offset of this axis from the rear wheels. The latter determines the 'feel' or 'behavior' of steering as discussed above. This feature also greatly enhances the maneuverability of the vehicle.

A2.2. Unlimited Steering

In the absence of mechanical connectors between wheels, it is possible to design a steering mechanism for unrestricted turning of the wheels. Figure A2.2 illustrates a conceptual design for the unrestricted steering of a slow moving vehicle. The wheel is mounted on a long bracket which can spin around the vertical axis. This bracket has a gear on top that engages the pinion of a servomotor. The figure shows this gear and the servomotor's mounting holes on the chassis, but it does not show the servomotor itself.

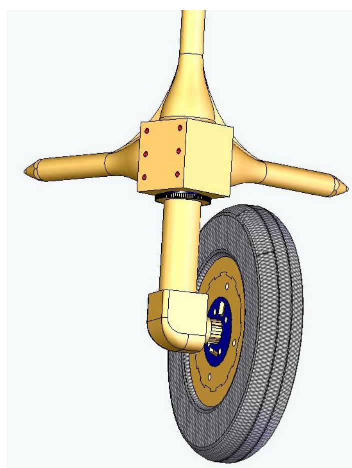


Figure A2. 16: Unrestricted turning of the wheel.

With such unrestricted steering as the one provided by the above design, the distance between the ICR and centerline can be made as small as zero. This enables the vehicle to turn in a very small radius, or even turn around on the spot.

A2.3. Adjustable Sensitivity

Here the term 'sensitivity' refers to the ratio between the driver's steering and the actual turning of the wheels; high sensitivity means that for a small turning of the steering wheel the vehicle turns to a large degree. As a safety feature, it might be desirable to vary the sensitivity of steering adversely proportional to the 'speed' of the vehicle; as a performance feature, it might be desirable to vary the sensitivity of steering for different turning radii. Figure A2.3 illustrates this adjustability.

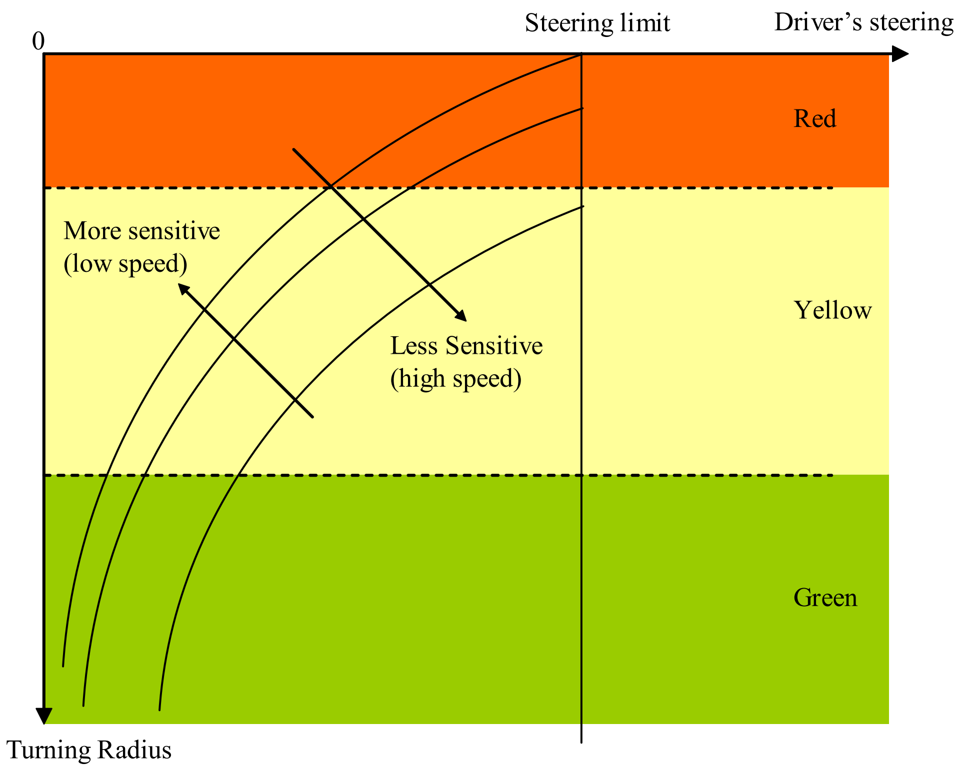


Figure A2. 17: Adjustable sensitivity in servo steering.

The horizontal axis on top is the driver's steering, ranging from zero to the device limit. For example, if a steering wheel is used as the driver's interface this limit might be 720 degrees (two whole turns). The vertical axis on the left is the turning radius that varies from infinity to a minimum, which in this design can be as small as zero. The figure shows three sensitivity curves for three different speeds. Also, the steering radii are color coded. Red represents a radius smaller than half of the wheelbase, which means the vehicle spins around itself and the inner wheels have to move in the reverse direction of the outer wheels. The green zone represents larger thus safer radii, and the yellow zone represents smaller radii which are still large enough to keep all four wheels moving in the same direction. It can be seen that as the speed increases, the sensitivity is reduced. The high speed curve on the right requires more steering to achieve the same radius that would be achieved by a small steering at low speed. Further, it can be seen that the high speed curves do not reach the red zone. These curves are examples of many performance profiles which can be easily programmed for the servo steering system.

A2.4. Wheel Alignment

Wheel alignment in a conventional steering mechanism assures that at ‘zero steering’ all four wheels are paralleled along the vehicle’s centerline. Once this is achieved, the *differential rotation* of right and left wheels is automatically achieved through the geometry of steering knuckles. In the servo steering mechanism proposed here, wheel alignment means both the *calibration* of servo motors and the *resetting* of ‘zero rotation’ for straight driving.

The resetting process is similar to the process of conventional wheel alignment. All wheels are paralleled along the centerline and the *reference point* (zero) for servo motors and for their associated *feedback sensors* is reset to zero. The calibration process involves the ‘training’ of servo motors in order to achieve the desired sensitivity discussed in Section A2.3. Both resetting and calibration can be performed easily because these processes only require the adjustment of *software* variables.

Appendix 3. Task and Information Processing

Chapter 4 defined a *task* as a process that begins with a set of goals and ends when these goals are achieved according to the task's *design*. Design is a plan, recipe, or a set of instructions for performing a task. Execution of a design requires *work*. This thesis uses the term *information processing* (IP) to refer to the 'work' or 'effort' spent to execute a design. The reason for this terminology is that any work can be theoretically translated into the execution of *primitive instructions*.

A task has a hierarchical structure, and is always decomposed into smaller tasks. As explained in Section 4.1, the decomposition ends arbitrarily depending on which detailed levels of subtasks are pertinent to the task at hand. For example, the decomposition of a task may continue to extremely detailed levels such as the movements of individual *basic particles*, if such details are important. However, such level of detail is rarely relevant to an engineering task. A reasonable end-level of decomposition detail for engineering tasks might be to treat the human body with all its functionalities and capabilities as an available resource. In this case, the decomposition may end when the design is broken down into a set of "primitive instructions" (e.g. neural pulses) that will create the required body motions to finish the task.

In practice, however, this level of detail is unnecessary. Decompositions actually end at much higher levels than individual body movements. Humans can learn procedures and skills, and can on their own plan the required course of action for simple tasks. Thus these skills may be treated as a higher-level available resource; these resources can then internally generate the required primitive instructions.

Chapter 4 explained that a design (set of instructions) can be treated as an *end-node* only if its fulfillment can be assured by the available resources and thus does not require further decomposition by the designer. The above discussion reveals that such resources are available at different levels and are capable of performing tasks at different levels of complexity. For example, a manufacturing firm can be treated as an available resource. Once the manufacturing drawings are given to this entity, the delivery of the physical product is reasonably assured, and there is no need for the designer to decompose the task and specify the details of production processes.

Figure A3.1 shows the hierarchical structure of technological resources discussed above. A '*task robot*' in the figure is an emulation of a human body with the same physical abilities. It requires detailed instructions (or programming) in order to perform a simple function such as cutting a metal bar. The middle part of the figure shows a higher level resource, which includes a set of skills. In this case, it is unnecessary for the task to be decomposed into detailed instructions. Skilled resources have the ability to process information and prepare the detailed instructions for task robots from a set of high level instructions.

The bottom of the figure shows a higher level resource, which is a *service provider* such as a company. A service provider internally consists of a hierarchical structure of lower level resources such as labor, skills, machine power, and computing power. A task can be assigned to these service providers at high levels of complexity, while we can be reasonably certain of the outcome. These service providers internally do the *information processing* and generate the set of lower-level instructions for the task. That is, they internally decompose the task into detailed instructions for their subservient entities, and allocate resources accordingly.

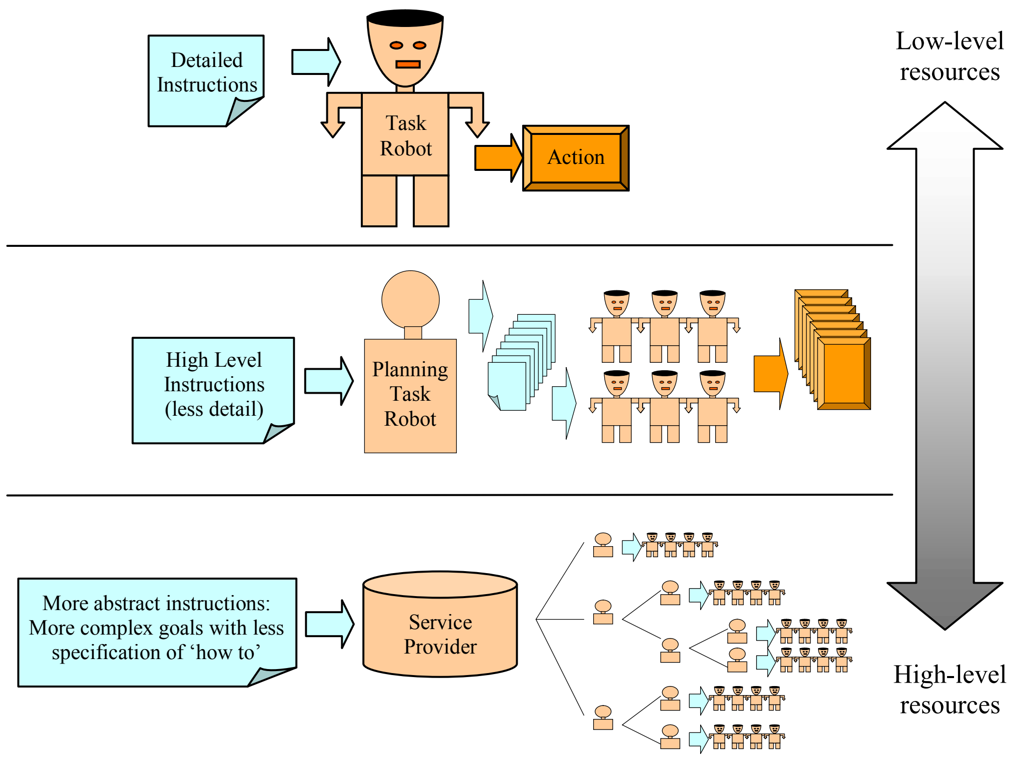


Figure A3. 18: Hierarchy of service providers, the resource for high-level functions.

This discussion reveals that accomplishing a task, regardless of its size, involves the processing of information and ultimately decomposing the task into a set of "primitive instructions". Thus, "working" is in fact "information processing", and the amount of work can be understood as the number of "primitive instructions" which need to be executed. In summary:

- A design is a plan or course of action for achieving a goal.
- A design can be ultimately translated into a number of "primitive instructions".

- The “information content” of a design is in fact a “price tag” which is attached to it. This tag indicates how many primitive instructions have to be executed in order for the design to materialize.