ENHANCING PRODUCT DEVELOPMENT THROUGH COLLECTIVE SYSTEM DESIGN DECISIONS AND SUPPORT

A Thesis

Submitted to the Faculty

of

Purdue University

by

Anderson E. Miles

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Engineering

May 2020

Purdue University

Fort Wayne, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. David S. Cochran, Chair

Professor of Systems Engineering and Director, Center of Excellence in Systems Engineering, Purdue Fort Wayne

Dr. Zhuming Bi

Professor of Mechanical Engineering, Purdue Fort Wayne

Dr. Yanfei Liu, Ph.D

Assistant Professor of Electrical and Computer Engineering, Purdue Fort Wayne

Approved by:

Dr. David S. Cochran

Professor of Systems Engineering and Director, Center of Excellence in Systems Engineering, Purdue University Fort Wayne

ACKNOWLEDGMENTS

First, I want to thank my wife Xi Luo, mom Linda Cobb, and dad Jeffrey Cobb for supporting me through out all of my life adventures. Without their support and devotion, I would not have been able to pursue my dreams in education, travel, and general happiness.

I want to thank Professor David S. Cochran, John Fitch and Joseph Smith for their support and guidance during this academic pursuit. They have always challenged me to be more thorough in my research and writing. Due to their diligent support and patience my ability as researcher and writer has grown tremendously.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

SYMBOLS

PFW	Purdue Fort Wayne
SE	Systems Engineering
CSD	Collective System Design
AD	Axiomatic Design
FR	Functional Requirement
PS	Physical Solution
FRm	Measure for Functional Requirement
PSm	Measure for Physical Solution
PDCA	Plan-Do-Check-Act

ABSTRACT

Miles, Anderson E. MSE, Purdue University, May 2020. Enhancing Product Development through Collective System Design Decisions and Support. Major Professor: David S. Cochran.

Design is complex and an important way to deal with the difficulties of design is to meet designer needs within the product development process. Many design methodologies are inapplicable and inappropriate in design practice because they were developed to be context-free universal processes. This thesis argues that design involves social and technical context. The social context influences the design process through social factors like negotiation. Negotiation will determine the outcome of major design decisions. Therefor the design process should guide designers in addressing social factors. The technical context influences design through how the design requirements are linked with the non-static environment in which design takes place. Therefor the design process should guide the designer in addressing the technical context as each tool or design methodology will determine the very character of the solution space. This thesis will identify the social and technical designer needs. This Thesis then builds on the core concepts of Collective System Design and Decision Drive Design to create a guideline for conceptual product design. This thesis then applies the guideline to develop a product to show that meeting designer needs will improve conceptual product development.

CHAPTER 1. INTRODUCTION

1.1 Research Objectives

This thesis is comprised of one research question and three objectives. What are the designer needs within conceptual product development? The first research objective is to characterize design support and identify designer needs that should be addressed within the design process. The second objective is to propose a design guide to address designers needs within the conceptual product development process. This research objective is used to show that building upon two known methods, Collective System Design and Decision Driven Design, will meet designer needs. The third research objective is to apply the proposed design guide, developed in research objective two, to develop a product that would enable transradial amputee patients to regain normal hand functionality. The purpose of the third research objective is to validate that the integrated method can describe a new and innovative product that will improve the quality of life of people.

1.2 Outline of Thesis

This thesis begins by defining what is meant by a system and the function of system engineering. This introduction concludes with defining the research hypothesis.

Chapter 2 provides a literature review of system engineering approaches used today. The topics covered are Waterfall, V-model and Agile. Each method approaches the design process differently and as a result has an array of strengths and weaknesses.

Chapter 3 addresses the first research objective by providing a discussion on the complexity and fundamental difficulties of design. This chapter works to describe social and technical context within the design process to identify design process needs to support the design team.

Chapter 4 provides an introduction to Collective system design and Decision Driven Design. The core concepts from both design methodologies will be used in chapter 5 to propose a guideline for conceptual product design.

Chapter 5 covers the second research objective by proposing a design process to meet all social and technical needs. The proposed process is an integration of Collective System Design and Decision Driven Design.

Chapter 6 covers the third research objective by applying the proposed design process to design an electric prosthetic hand system to enable transradial amputee patients to regain normal hand functionality.

Chapter 7 provides a summary of the thesis along with further research objectives and suggestions.

1.3 Introduction to Systems and Systems Engineering

The word "system," is generally described as the interactions between system elements towards a common objective [1]. There are many ways to define systems engineering. For the purposes of this thesis, the following two definitions will be used: Systems engineering is the integrated approach that enables the creation, use, and management of an engineering system through the use of system principles, technology and tools [2]. The key word is *integrated* as individual system elements by themselves do not constitute a system. The second definition is as follows: The function of systems engineering is to guide the engineering of complex systems [1]. The key word being *to guide*, which emphasizes the process for the selection of a path for others to follow. There are many design methods or tools used within the systems engineering field such as Axiomatic Design, Collective System Design (CSD), Agile, Decision Driven Design, Design Thinking, and many more. Collective System Design, Agile, and Decision Driven Design all have various strengths and weaknesses in design and will be used to build an integrated design approach discussed within this thesis.

1.4 Design Decision Making

Design is a creative process with purpose, which inevitably consumes resources. Therefore, design can be characterized and examined. All design problems are subject to certain criteria that enable the designer to decide which response is a better solution among alternatives. As a result a design problem, at its core, is a decision-making problem. Major difficulties in design arise in the handling of mass information in order to reach sufficient results. unlike algebra problems, many feasible solutions exist that possess differing characteristics and consequences. The primary purpose of design theories is to establish design confidence in the face of highly dependent external factors in which many solutions are difficult to evaluate in advance.

The use of models and traditional design methods play an essential role in communication and learning. However, many traditional design methods are not adequate in meeting all of the customer's needs and designer needs in new product development. Traditional methods are not adequate due to the increase in overlapping problem domains and time constraints. In other words designers are expected develop products with more features with increasing complexity and in shorter times. Providing appropriate design support is the solution and contains both social context and technical context. Social context refers to how design methodology enables a development team to work together. Technical context refers to how the process facilitates turning design ideas into a design commitment. This thesis argues that any design methodology should provide adequate design support to meet both end user needs and designer process needs.

1.5 Hypothesis

The hypothesis of this thesis (H_A) is that meeting designer needs will improve conceptual product design. The null hypothesis H_0 is that conceptual product development will not be improved by meeting designer needs. The main objective of this thesis to provide a logical argument to reject the null hypothesis in favor of the H_A. This thesis uses information modeling to attempt to provide the burden of proof as to why the null hypothesis can be rejected in favor of H_A.

H_0: Meeting designer needs in Conceptual product development processes will not improve the product design.

H_A: Meeting designer needs in Conceptual product development processes will improve the product design

CHAPTER 2. LITERATURE REVIEW

This chapter provides an overview of two product design methodologies and is divided into two sections. Section 2.1 will focus on the following: principles of design and their associated activities. Section 2.2 identifies the strengths and weaknesses of two commonly used design methodologies which establishes two sides of a spectrum of how design methods handle technical management processes. The contrast between the two methods raises the following question: "Is it possible to have a stable, predictable process, but also quickly respond well to change?" This question sets the stage for characterizing design to identify design needs within the conceptual product development process described in chapter 3.

2.1 Definition and Purpose of Design Theory

There are many definitions of a theory and most can be summarized in the following quote: “the theory should describe and provide a foundation for explaining and predicting the behavior of the concept or object” [3]. For the rest of this thesis, design theory is a body of knowledge that is used to describe, predict, and explain concepts. Koskela offers a view on the purpose of design theories. The theory should improve or provide better “explanation, prediction, direction and testing” and “provide tools for decision and control, communication, learning and transfer” [4]. In much of the current literature the aims of design theories are to create support for the designer and ultimately to improve understanding of information obtained and to improve the way information is used. Design theory not only aims to characterize design but also to simplify the design process. Many authors attempt to simplify the design process through models and theories but do not distinguish the difference between the two concepts [5]. A model and design theory both have a clearly defined boundary of what it can describe and should simplify complex problems [6]. While design theory and models both aim to understand design through constructs and relationships, models have the following three distinct characteristics:

- Attributes of original information is transformed into model attributes
- Reduction of attributes from original information to model information
- Consist of pragmatic characteristics

This thesis argues the need for integration of models to aid in the design process. To obtain an understanding of how support for a development team within the design process should be expressed, it is necessary to understand how product development methodologies approach design. The following section provides a brief overview of the design processes within product development.

2.1.1 General Technical Processes

There are three sets of technical processes which are common across product development systems. The three processes are system design, product realization, and technical management [7]. The system design process is used to frame the project through stakeholders and define system requirements and design solutions [7]. A stakeholder is an individual having a claim on or in possession of system characteristics that meet their needs and expectations [8]. Technical requirements are statements about what the system must achieve and are derived from stakeholder expectations [1, 7]. Once the technical requirements are identified, a logical decomposition must be completed. A logical decomposition is the process by which requirements are allocated into a set of models for the input of design solutions. Each requirement is analyzed by time, behavior, data flow, and object to form criteria to define the appropriate solution space. After the solution space is defined, the design solution is picked. The design solution is the physical entity which will meet a system need.

The system design process is repeated until the full system is realized. To manage interactions between systems elements, a system management process is required.

2.1.2 Technical management process

The goal of the technical management process is to provide bidirectional traceability and manage changes to establish requirement baselines [7]. A technical management process has the following two components: interface management and risk management. An interface is a functional or physical connection between two or more components with the system [9]. Interface management is the process that identifies, controls changes, and documents interfaces. Interface management must meet the following requirements [7, 9]:

- Enable system elements to co-function by providing necessary design information
- Ensure sustainable uses of interfaces through the use of interface performance criteria and constraints

Every decision made within the design process exhibits some associated risk. Risk is a measure of potential failure in any system objective such as cost, schedule, and technical constraints [10]. The goal of risk management processes (RMP) is to identify potential pitfalls and risks in the following areas [11]:

- Technology risk
- Risk mitigation strategy
- Contingency planning

There are many ways to approach design and many methods claim to include the necessary design principles and activities to ensure project success. Many projects apply basic design principles within design processes but still may fail to meet project goals. A common question in project planning: should risk management processes (RMP) be implemented earlier or later in the product life cycle? Guidelines correlated with impacts of implementing RMP earlier versus later in the project life cycle present many complex issues. Waterfall and Agile are examples of product design methods which seek to aid design teams through establishing a framework for the design process. Waterfall, V-model and Agile methods differ strikingly in their approach to product design and offer distinct strengths and weaknesses.

2.2 Current Product Development methods

Much of the literature suggests that most of industry is using Waterfall or Agile base product development methods. Each method approaches the development process differently and has many strengths and weaknesses.

2.2.1 Waterfall and V

Winston Royce, a computer scientist, introduced the Waterfall Method in 1970 [12]. Waterfall provides easy usability due to the rigidity of the model. Each phase within the Waterfall Method does not overlap and are processed and completed one at a time. Traditional methods, such as the Waterfall model, suggest a formal separation between project planning and risk management planning. Fig. 2.1 and Fig. 2.2 depict the Waterfall model and V-model.

2.2.2 Agile

Agile "scrum" tackles product development through iterative processes. The focus of Scrum is to handle the dynamics of product development through fast validation feedback and client involvement. Scrum puts the business value functions into the hands of the stakeholder. The goal of client involvement is to allow the client to guide how the product is developed. The values gained from using scrum are the following:

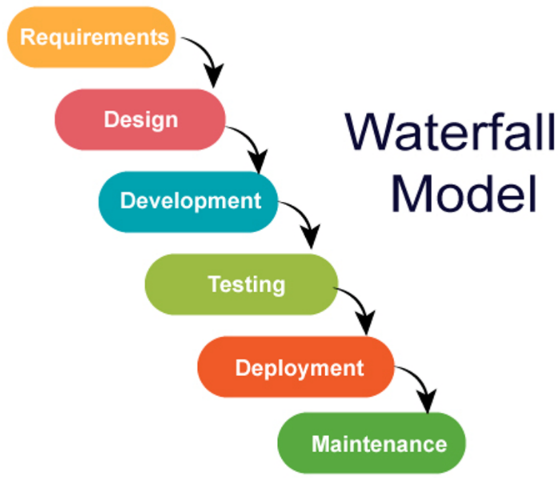

Fig. 2.1. Waterfall model is a sequential design process in which each phase does not overlap. [13]

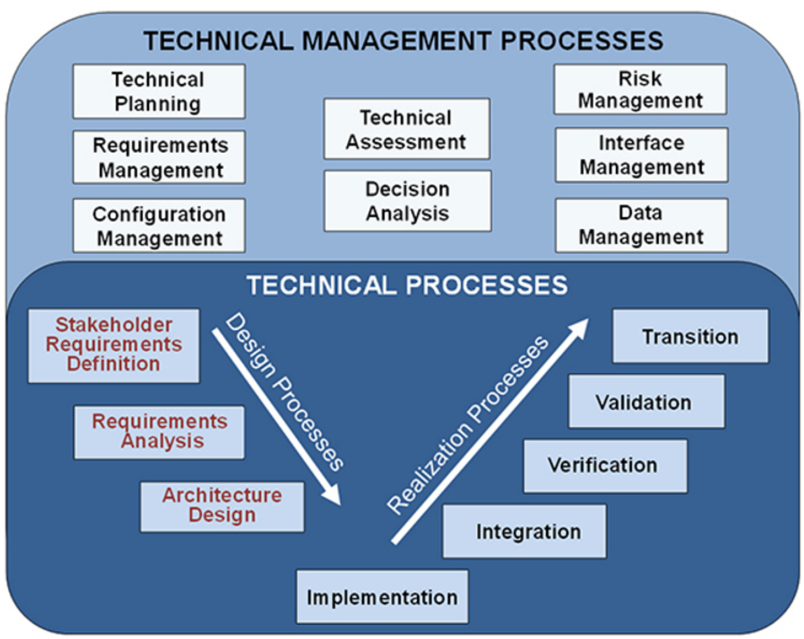

Fig. 2.2. V-Model is a sequential design process that separates requirement planning and validation planning. [14]

- Rapid feedback from users increases usability and quality of application
- Increased ownership of product owners
- Regularly meets specific measurable objectives

Scrum values emphasize interactions over tools and documentation, which enables design processes to be more responsive in changing environments [15]. Through each sprint a project team produces fully functional, tested and documented solutions. The major strength of Scrum comes from the integration of project planning and requirement validation through the use of sprints. The Scrum process is iterative and reappraises projects in each step of product development, which is likely to reduce overall project risk.

The major strengths of Agile are also the major weaknesses. According to Highsmith and Cockburn, Agile has a few disadvantages and presents the following issues [15]:

- Low visibility over the project outside of sprints
- Client involvement can damage project due to unclear sense of direction

Waterfall and Agile present two approaches to product development. Waterfall is predictable as this method front-loads much of the project planning but typically validates solutions in batches or later in the product design cycle. On the other hand, Agile is highly responsive to change as the method validates solutions in an incremental fashion, but often fails to scale up to large projects due to lack of a holistic system view. Table 2.1 summarizes the strengths and weaknesses of Waterfall and Agile methods [16].

Is it possible to organize and design all of the elements by the application of knowledge and test preliminary designs as soon as possible to validate solutions? This thesis suggests an integrated method which enables a design team to establish predictability and stability while validating requirements and implementing solutions as soon as

Table 2.1.

Agile VS. Waterfall

Project Characteristics	Agile	V /Waterfall
Application		
Primary goals	Rapid value, resonding to change	Predictiability, stability
Size	Smaller teams and projects	Larger teams and projects
Management		
Customer relations	Dedicated onsite customers	As-needed customer interactions
Planning and control	Internalized plans	Documented plans
Communication	Tacit interpersonal knowledge	Explicit documented knowledge
Technical		
Requirements	Prioritized informal stories and test cases	Formalized project, capability, interface
Development	Simple design, short increments	Extensive design, longer increments
Test	Executable test cases define requirements, testing	Documented test plans and procedures

possible. Recognition of design principles and common technical management processes may not be enough to ensure project success. With the plethora of methods available to develop products, why is design difficult? Chapter 3 will provide discussion on the social and technical contextual influences on the design process. Chapter 3 will also identify design support needs and develop a model to illustrate how design support should function.

CHAPTER 3. DESIGN CHARACTERIZATION AND SUPPORT

3.1 Introduction

Chapter 2 reviewed waterfall and agile product design methods and identified their respective strengths and weaknesses. Each method approached the technical management process very differently. The waterfall method focused on upfront planning and batch risk assessment and implementation. Agile focused on fast feedback with client involvement. Chapter 2 established that recognition of design principles and technical management processes may not enough to ensure project success. Chapter 3 is divided into the following two parts: Characterizing the design process context and identifying design support requirements. Characterizing the design process context sets guidelines for identifying design support needs. Identifying the design needs will enable the synthesis of the integrated process discussed in chapter 4.

3.1.1 Characterizing Design

The complexity of design has been apparent for many years. Product development teams are asked to include more functions and features in products for less cost and in less time. The common techniques for decomposing design problems in industry are the following: A division concerned with product life cycle and division of domains or disciplines. Both techniques have different approaches and subsequent consequences on design practice. Neither of the techniques alone are sufficient to explain how design should be approached. The different phases of the product design life cycle govern the extent and scope of different methodologies. A product life cycle generally consists of the following five phases [17]:

- Design
- Production/ Manufacturing
- Sales/ Marketing
- Service
- Product liquidation or recycling

This thesis focuses on the conceptual design process and addresses how to effectively meet both designer and consumer needs. The design process generally has three design phases [17].

- Product characterization: A statement of a product's specification considering sales as well as both technical and economic feasibility
- Product development: A plan of how to implement deliverables of the product
- Manufacturing/ Production: The method used to produce the product

With product characterization, product development is typically approached using an ancient technique of *divide et impera* (divide and rule) [7,9,18]. In most methods, the technique can be seen as the decomposition of a problem. Each design need is broken into modules until the need can be tackled by individuals working in particular engineering disciplines. The development team is then comprised of a variety specialists to tackle domain-specific problems. Decomposition of each need results in functionality within single domains or subsystems. The decomposition of domain specific needs results in a design process in which one sub system is designed after the other. This segregated design approach can reduce the design difficulty in simple product designs where the subsystems and their interfaces are clearly understood.

The above standard pattern approach may reduce some uncertainty of the design but presents many issues. First, most designers only regard solutions within perspectives of their sub-problem and as a result, the design phases becomes a "throwing over the wall" process [36]. This process is where the designers finish their part of a project and hand off their part to the next person with little to no communication. Second, design of subsystems without consideration of mutual interactions will at best lead to an overly complex design and at worst fail.

This thesis argues that conceptual product development requires social and technical context are addressed within the design process. Ineffectively addressing social and technical context will lead to product development that is costly and will cause more rework that could have been prevented. To provide sufficient support for the designer, both social and technical needs have to be met. To provide insight on why design support is needed, the following section provides discussion on the complexity of common issues in design.

3.1.2 Complexity of Design

The complexity of design is due to the evolutionary nature and interaction between design knowledge and design activities [19]. Both design knowledge and activities are based upon design artifacts and their respective audiences. Design artifacts are the collection of descriptions that include an audience and have a purpose. Examples of audiences are the following:

- Stakeholder
- Developer
- Product owner

Each design artifact has an audience and each audience has a purpose. Some examples of purposes are the following:

- Understanding the interactions with a system
- Using the product
- Providing guidance

Design audience, purpose, and the expression of artifacts are all parts of design knowledge. The design audience and purpose evolve from first concept, through modified design specification, to completion of design. Design knowledge leverages and activates design processes and includes knowledge that informs design and constrains the available solution space. Due to the evolutionary nature of design, the designer is always attempting to hit a moving target within the problem and solution spaces. Design artifacts are concerned with adapting to the evolutionary nature of the audience and purpose. This concern implies less focus on the question of "what is" and more emphasis on "what ought to be." Within natural science domains, experimentation is a common method for realizing necessary design artifact changes. Repeated experimentation is limited in product development due to cost and time constraints. Design support within the product development process should decrease the reliance on ad hoc processes and align an organization's needs, capability with cost and time constraints. This thesis argues that to identify designer needs, design must be viewed as a contextually dependent, evolutionary process.

3.2 Social Context

From the start, design problem descriptions are usually incomplete. The initial problem statement does not describe all of what a system must accomplish. As a consequence, each proposed solution may not be applicable to the problem domain and may even obscure or hide design issues. Each solution can only be considered valid or invalid after meeting the problem domain's criteria for success.

At the start of the design process, many alternatives are proposed and are considered equally valid in concept. Through each selected solution, the knowledge of the design view should grow while the design freedom, due to constraints, should shrink [20]. Each identified need should lead to an observed set of constraints that forms a point of view or frame and then the solutions can be proposed and tested.

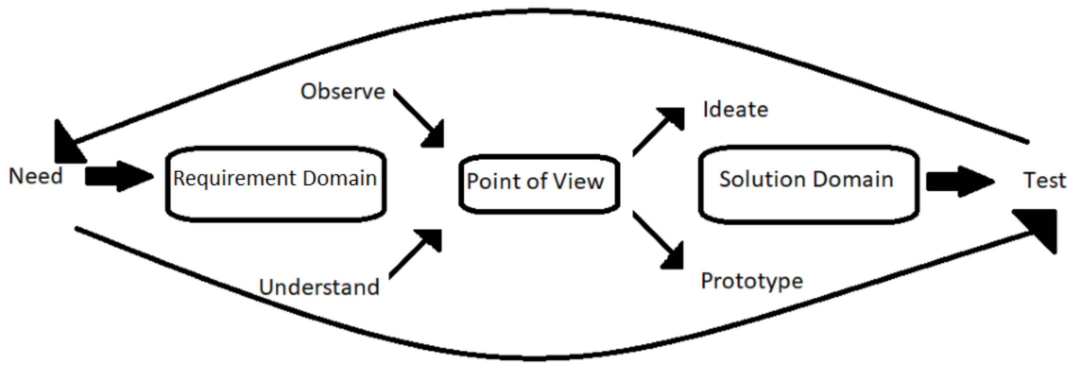

Fig. 3.1. Design Frame Process in which the designer formes a point of view to describe needs and test solutions

The need is an input to the requirement domain and the designer observes and attempts to understand by forming a point of view. From the designers point of view, they attempt to find a solution through ideation and prototyping. Each solution selected and tested can generate new needs. Figure 3.1 illustrates the process. Proposing a solution means to understand or (re)formulate the problem domain as each solution has its own derived needs and requirements. Figure 3.1 illustrates this process. The author argues that part of the difficulty in proposing an appropriate solution lies in the social context embedded within design. As most design occurs in teams, it is of utmost importance to ensure every team member shares a common point of view. A common point of view ensures that every designer is working toward addressing common system needs.

3.2.1 Social Context Needs

Empirical evidence shows that successful teams cycle through periods of high and low shared understanding and in the later stages of product development have a high shared understanding [21]. At the start of early product development phases, each team member has their own assumptions, or mental frames, which govern their actions and interpretations. Frames within product development have many meanings but can summarized by the following definition: Frames form the basis upon which the designers select and form requirement-solution sets [22] [23]. Individual frames become shared frames when the individual frames overlap or align. Due to the unique social context or background, which forms an individual's frame, negotiation within development teams is necessary in order to obtain a shared frame of the project. Successful negotiations within designs teams typically begin with addressing conflict between individual frames and end with high shared understanding [21] [24]. Conflict enables the negotiation of shared frames. The lack of shared meaning in common language, coordination and collaboration, makes obtaining a shared frame difficult in design teams. It would seem conflicts or misalignment in individual frames are beneficial, but only when conflicts are made explicit. The design process should facilitate shared communication and agreement of the following six social product development needs:

- Desired end state or goal - Each designer on the team share common overall end states or goals.

- Relative significance of product features- Each designer should know the hierarchy of product features.

- Boundaries to the design problem - Each designer should know the system boundary.

- Criteria for evaluating input and output information - Each designer should have a common plan on how to evaluate input and output information.

- Distribution of design responsibilities without loss of holistic view - The design responsibilities should be split, but each designer should retain a view of the entire system and its requirements.

- Manage uncertainty throughout design - The development team should have a plan and process to manage decision and design uncertainty.

By establishing a shared agreement of these six social design needs, the development team's understanding will be made explicit and thus allow a common frame to be formed where the design team shares common knowledge of design goals and available solutions. Ideally, the design team should be able to determine the best course of action by drawing and reflecting upon all of their knowledge [25]. Due to the individuality of each designer's frame, one should ask the following questions:

1. "What activities support communication toward forming a shared frame in the informal early phases of design"

2. "What technical activities establish and facilitate communication of the six social product development needs"

Each individual's frame is constantly evolving or diverging from other individual team members. it is challenging to obtain a common shared frame and one may never be achieved. This thesis argues that establishment of a common frame is vital for positive development process outcomes.

3.3 Technical Context

There are many design methods that have been used over the last few decades. The next section will discuss existing models on the basis of a taxonomy, instead of treating each method individually. The aspects that impact technical activities within design are purpose, modeling approach and how time is managed. The aim is to present a general model that explains the applicability of design support within technical context and to explore what technical activities support social context.

3.3.1 Technical Design Purpose

Design models used within technical activities can have the following models with regard to purpose:

- Descriptive

- Prescriptive

Descriptive models can be regarded as what actually is done during technical activities. Prescriptive models intend to describe how design should be done [26]. Descriptive models are required in all design activities as they serve to explain features within the design, and most importantly, identify interfaces between system elements. To capture interfaces effectively, an object-oriented approach is preferable, since components and their context differences change much less than operations or processes. Consider that historical methods for sharing information within systems engineering were document-based models. Document based models force engineers to do the following:

- Capture information in large free-form paragraphs
- Create and maintain unique requirement identifiers

Document-centric approaches typically fail to distinguish object and relationship data within large paragraphs [27]. This leads to high variability in technical processes and operations as differing interpretations among designers lead to further ambiguity within product design [27]. For example, when unexpected issues manifest within product development, the process model fails because the steps or procedures are no longer appropriate. In an object-oriented model the objects can adapt to new or redefined requirements if the unexpected event can be traceable to all system elements that it would affect. A precise method for capturing system engineering knowledge would be to define a descriptive object-oriented model. A model is proposed in the following sections that is descriptive and object-oriented. The goal of the model is to predict how design support should function rather than planning when design support is needed.

3.3.2 Developing the Model

Each design action will change the design state within product development. A design state is when the attributes or values of a system remain steady for a meaningful period of time [28]. The overall goal of the design process is to transform a concept into a description that can be used to manufacture the product. The process is depicted in figure 3.2 to show the that there are three different design blocks, from design concept, to the physical design action, to representative descriptions.

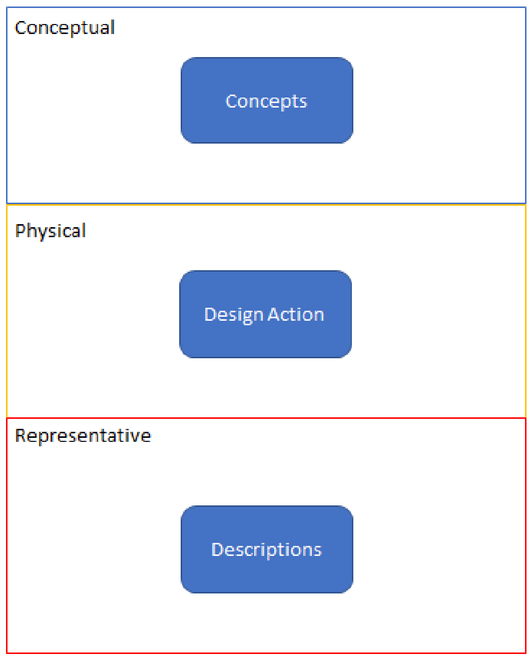

Fig. 3.2. General Design Process

Note that, in general, observations will require an activity to form a concept of descriptions, and turning concepts into a description will also require an activity. The concept and description design actions do not have a mutual interface as the physical world is always in between (activities and entities). To clarify the technical context, it is necessary to break design blocks into the following three components: design information, design representation and design process. Decomposing the design block will give further insight as to how design should function.

3.3.3 Refining the Model

There are many design methodologies that people use to simplify their product design processes. Some may not be conscious that each design methodology may have different purposes, and as a result, process information in dissimilar ways. All design methodologies have design actions and activities embedded in the process. Design actions and activities consist of the following: analysis, synthesis, and evaluation. Analysis is focused on capturing and transforming customer needs into system requirements. Synthesis is the process of generating possible solutions. Evaluation is how each solution meets each system requirement [7]. Each design block can process information in the following ways:

- Create - Forms new instance of information
- Select - Selects one entity over another
- Compare - Forms hierarchy of an entity
- Simulate - mimics and environment or entity
- Refine - Strips unneeded information
- Represent - Encompasses information such as document or test result
- Interpret - Explains or elaborates the meaning of information
- Constrain - Confines or restricts available options
- Test - verifies or validates information

Recognizing how a design methodology facilitates information processing is important to defining What design support a methodology will offer to a product development team.

3.3.4 Conceptual Block

This section will breakdown the conceptual design block that was illustrated in Figure 3.2. Concepts consist of needs, solutions concepts and the designer mental model. Both the needs and the solution concepts are captured within the designer mental model. Fig 3.3 illustrates the sub objects within the Conceptual block.

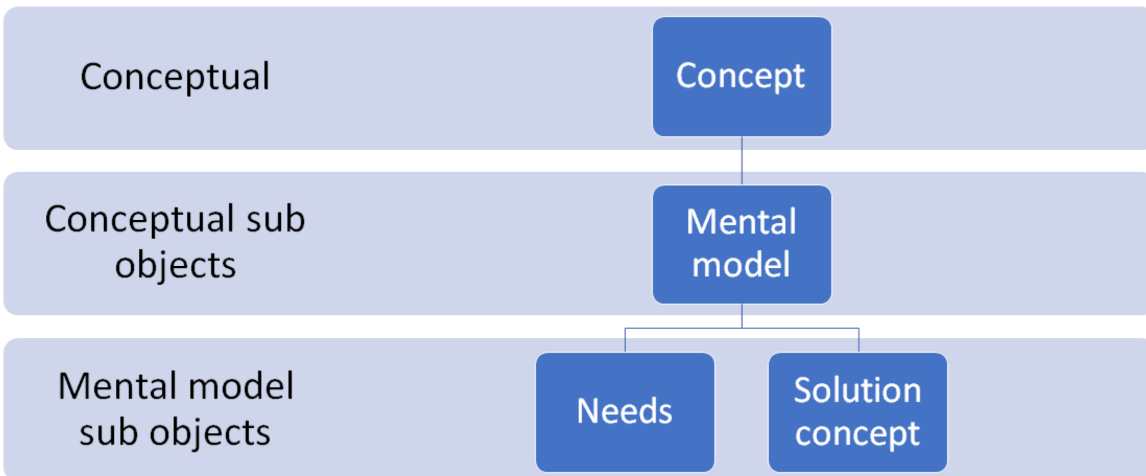

Fig. 3.3. Conceptual sub objects that break down the conceptual design block

The Design information contained within the conceptual block are the customer needs and the solution concepts and are represented within the designer mental model. The design process cycle is as follows:

- Needs constrains solution concepts
- Designer mental model represents both the needs and hypothesized solutions
- Designer interprets customer needs

The design information actions are contained within the conceptual block summarized in figure 3.4 below.

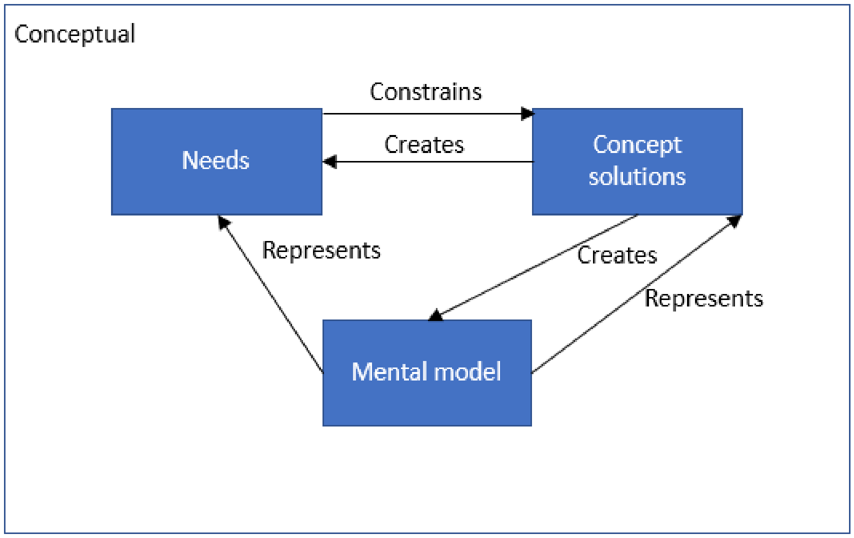

Fig. 3.4. Conceptual Information between each entity within the conceptual design block

The following analogy demonstrates the process above. A person named Jason wants to go on vacation in North America in February. Jason needs the vacation location to be warm. This need restricts the available vacation locations or concept solutions to only places that are warm in North America. One can say with confidence that Jason will not be vacationing in a place like Alaska and is more likely to pick a place like the state of Florida. The vacation example is similar to some design decisions in a way that a designer would not select a 1000 dollar component with a 500 dollar design budget. The need and solution concepts are both captured together in the vacationer's mind as he or she has not taken any actions yet and has made the first attempt of aligning needs with concept solutions.

3.3.5 Physical Block

This section will breakdown the physical design block that was illustrated in Figure 3.2. Informing and describing are two distinct processes of the physical world and should be defined from the viewpoint of the designer since he or she is responsible

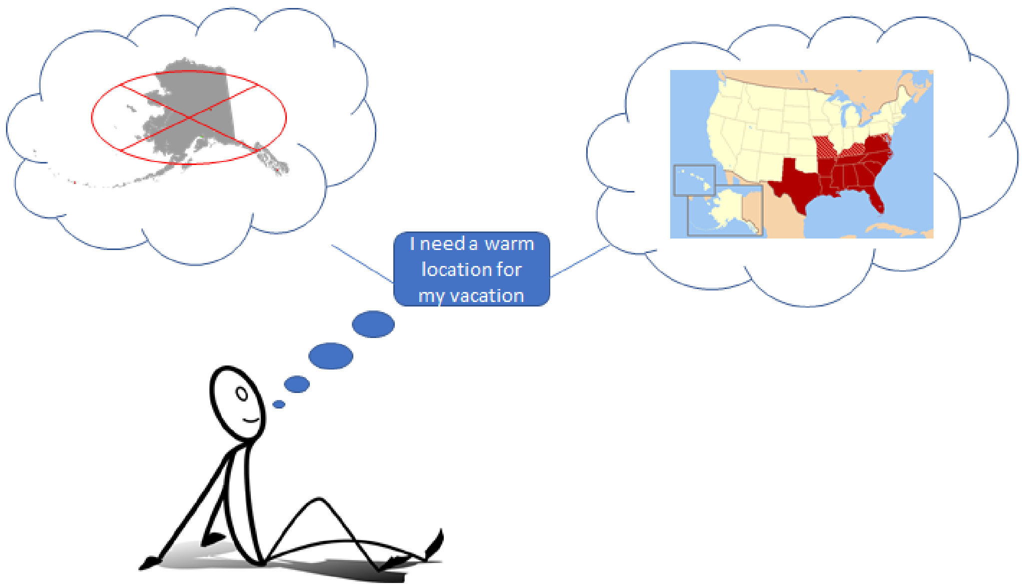

Fig. 3.5. Jason will not be vacationing in Alaska because the location does not meet his needs

for receiving and producing the design information. The designer has the functions of observing and deciding.

Each decision within the physical world will lead to an event within the design. The physical block is illustrated in Figure 3.6.

The design information within the physical block originates from the designer observations and events. Both information types are represented the decisions made. Designer decisions are the most important part of the process as nothing moves within product development without decisions and commitment. The process is as follows:

- Observations refine decisions.
- Decisions create events to move on to the next part of design or to gather more information.
- Observations interpret events.

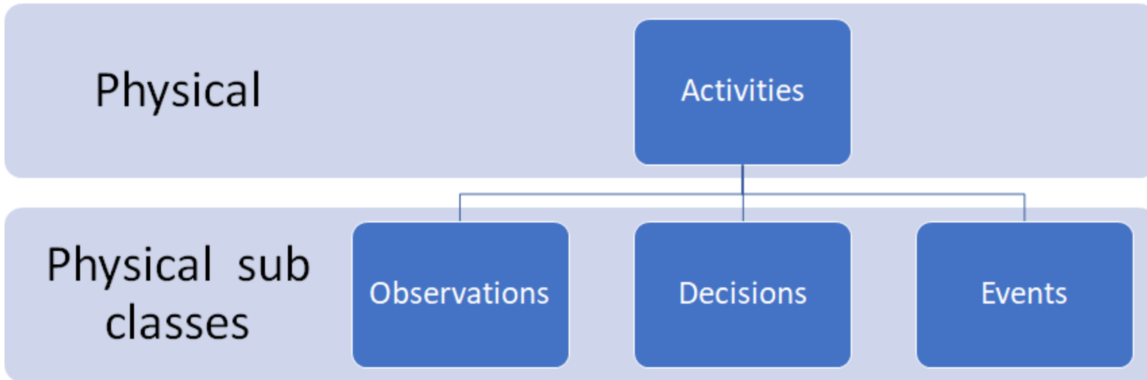

Fig. 3.6. Physical sub objects

The process within the physical block is illustrated in Figure 3.7.

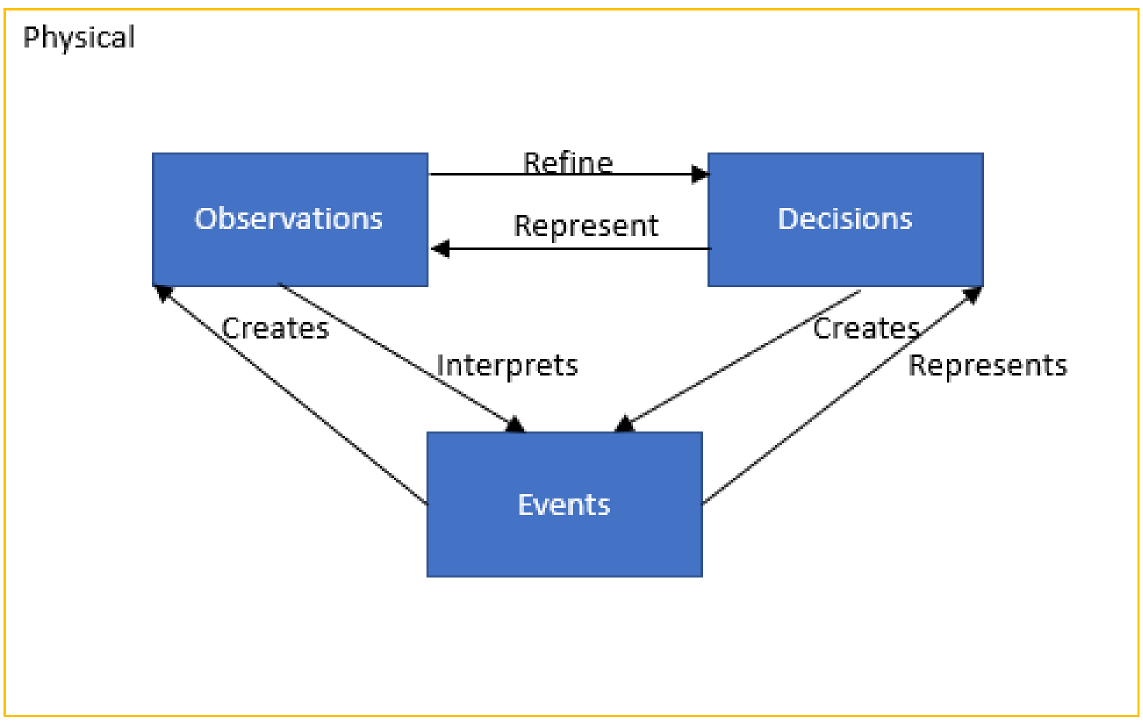

Fig. 3.7. Physical information

The design process is evolutionary in nature, from the beginning of projects to the end. The evolutionary nature within the physical world is the result of the numerous event options and the changing criteria of each decision. The complexity of which event to perform should always be dictated by the decision criteria.

The reason that events should be directed by decision criteria is that events cannot be initiated without the commitment of a decision to do so. The criteria for decisions are based on the level of confidence of the designer and information available. Events have following four distinct features:

- Level of confidence
- Quantifiable measures
- Level of resolution
- Numbers of alternatives

The level of confidence is considered to be the level of established correctness of a given solution as it evolves through the development process. At the start of a project the designers consider limited aspects of a design problem and are not specific concerning the implementation sequence of each aspect. Later the aspects are represented in more elaborate features and the freedom to define such features are limited but still belong to the aforementioned abstract aspects.

Quantifiable measures are considered to be the precision of detail of the design feature sets. In other words, the quantifiable measure is the exactness of the specified design attributes. For example, the designer will specify a range the specification must fall into, and only state exact quantities when needed.

The level of resolution is considered to be the amount of detail to be examined. The level of resolution changes throughout the design process. At the start of the design process, only the main features are considered, and the lower priority features are considered later. For example, in designing a watch, the main function would be the “display time of day” function. In addition, the function of “display heart rate” like an Apple watch, has a different scope and more issues are regarded. As a result, the structure of the features and their mutual dependencies changes during the design process.

Number of alternatives is the comparison of several equivalent solution sets to a problem. The designer selects the solution set most likely to succeed in solving the problem set. The number of alternatives considered differ in three ways: number of features, configuration dependencies and feature parameterization (quantification).

3.3.6 Representative Block

This section will breakdown the representation design block that was illustrated in Fig. 3.2. Design description within the representative block is the collection of descriptions and symbols that define design artifacts and design knowledge. Design knowledge refers to the technical design standards such as a building construction standard. The design artifact is the documentation that reflects the design model. The design artifact could be an implementation map, behavior diagram, frame, prototype etc. Figure 3.8 illustrates the sub objects of the design description in the Representative block. The design process within the representation block is illustrated in figure 3.9. The only function of the representative block is to capture what has been done within design and to pull information from common design knowledge.

3.3.7 Incorporating Conceptual, Physical, Representation Processes

This thesis has identified three fundamental blocks of product development: conceptual, physical and representational blocks. Two changes are needed to enhance the model. First, it is necessary to describe the process interactions between the conceptual, physical, and representational blocks. Second, the design process is almost always a team activity and the model does not currently address multiple designers. The next section will address the process interactions between the design blocks and

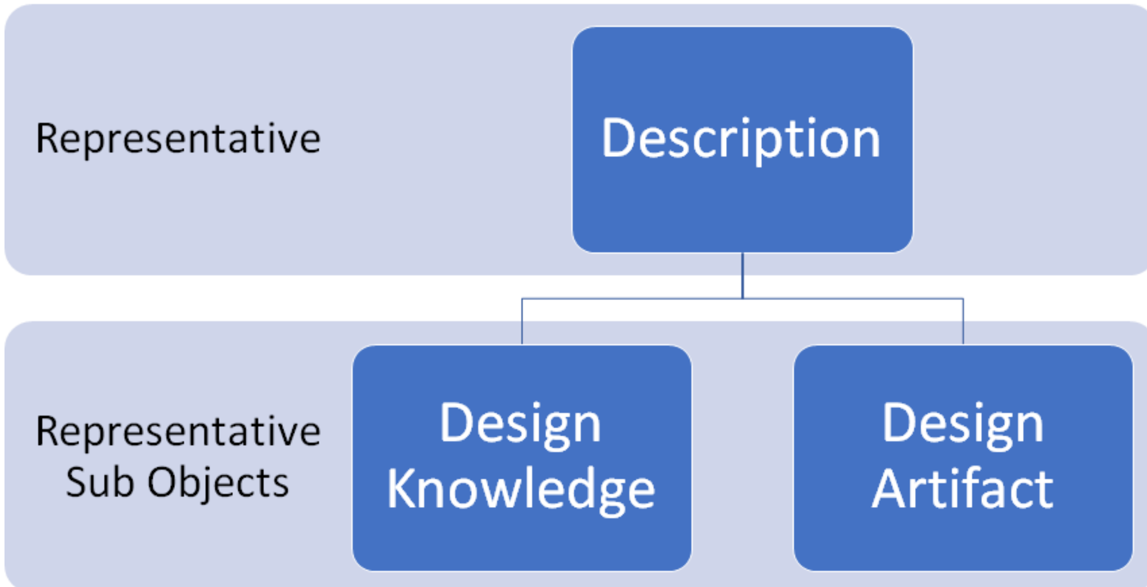

Fig. 3.8. Representation sub objects

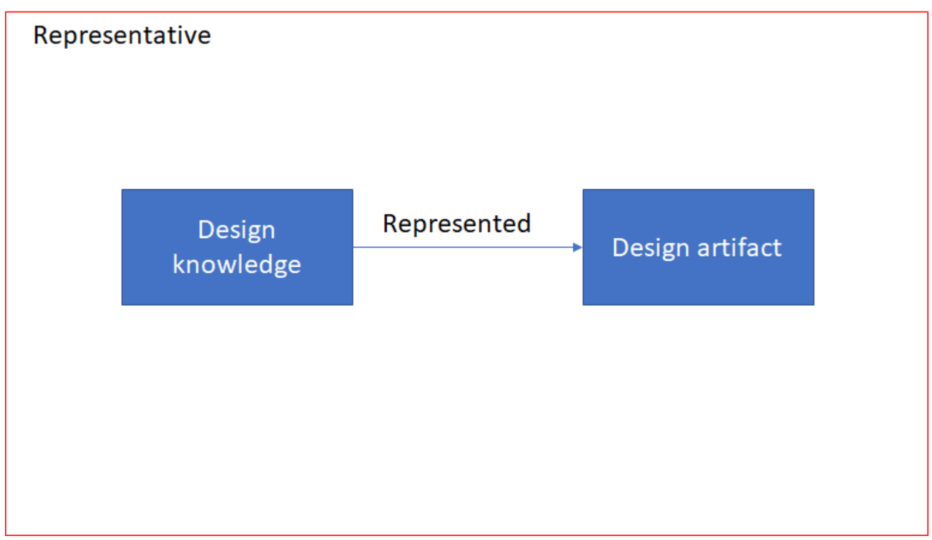

Fig. 3.9. Representative information

provide discussion of the application of the model within a design team. Table 3.1 represents the one-to-one interactions between each block. For example, the need in row 1 refines concepts as shown previously in Figure 3.4.

While there are many one-to-one interactions illustrated in the Table 3.3.7, there are many paths the designer can take to arrive at a complete solution. A complete solution is one that meets all the needs of all the stakeholders involved. Table 3.3.7 cannot illustrate all of the interactions and paths between each element within the model. However, decisions have the most one-to-one interactions. Decisions will compare needs, refine concepts, represent observations and the designer's mental model; as well as create events, design artifacts and design knowledge. Each decision is constrained by previously identified needs, and is refined by the designer's mental model and observations. This is illustrated in Figure 3.10.

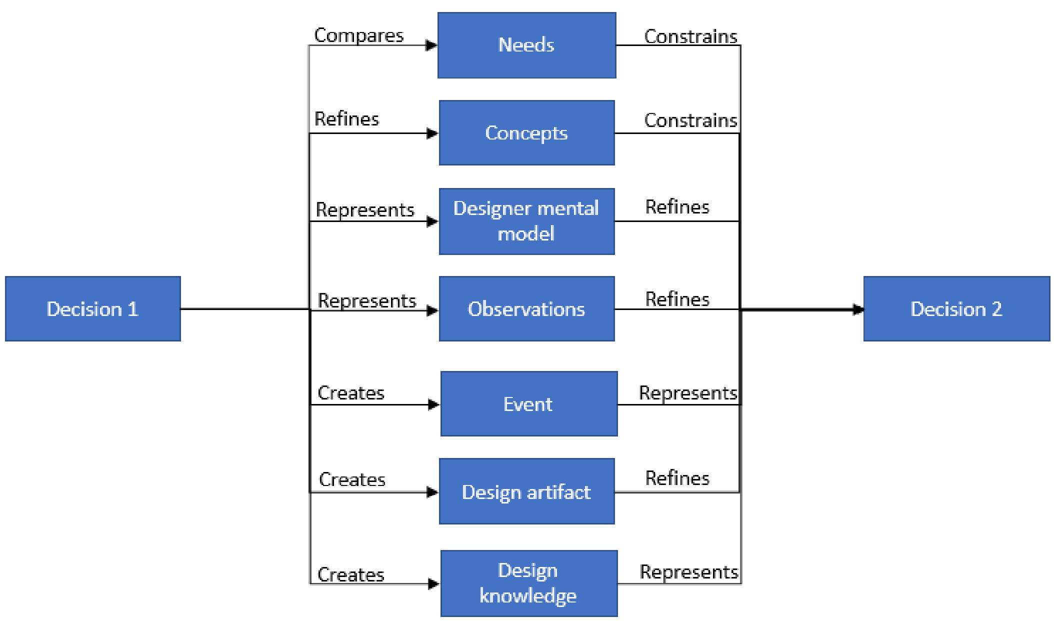

Fig. 3.10. Decision 1 forms the basis for inputs and constraints on decision 2. This process is repeated throughout entire design process

Table 3.1.

Design interactions

		A	B	C	D	E	F	G	H
		Needs	Concepts	Designer Mental Model	Observations	Decisions	Events	Design Knowl-edge	Design Artifact
1	Needs		Refines	Represents	Interprets	Constrains			Creates
2	Concepts	Creates		Creates		Constrains			Creates
3	Designer Mental Model	Represents	Represents		Interprets	Refines			Creates
4	Observations	Interprets	Compares	Refines		Refines	Interpret		Creates
5	Decisions	Compares	Refines	Represents	Represents		Creates		Creates
6	Events	Represents	Represents		Creates				Creates
7	Design Knowledge		Refines	Refines	Represents	Refines	Represents		
8	Design Artifact	Represents	Represents	Represents	Represents	Represents	Represents	Represents	

A decision-centric model will establish technical support (confidence) by meeting the following needs:

- Capture the source of knowledge within the design
- Support multiple solution concept analysis
- Create and accelerate convergence toward a solution
- Capture needs from multiple sources
- Decompose operational and system requirements
- Manage decision consequences
- Manage roadmap or plan

The reason for this is that decisions are both the creative and convergent process by which any designer will tackle problems and implement an appropriate solution. Decisions are the driving factor for how the problem-solution set is formed and executed. Any event or documentation that occurs within the design process is represented by the design artifact. The mutual interface between the conceptual and representation blocks is the decision actions made with the design process. Each commitment made to move forward in design must be represented by the needs and appropriate solution. The design teams must share a common agreement to problem frame and design decisions. If more information is needed to progress, events are created based on the decision criteria. The design team makes observations with the goal of establishing design confidence.

CHAPTER 4. COLLECTIVE SYSTEM DESIGN AND DECISION DRIVEN DESIGN INTRODUCTION

4.1 Introduction

This chapter provides an overview of Collective System Design (CSD) and Decision Driven Design (DDD). The objective to give background into each methodology and their key components. The two methodologies will be integrated in chapter 5 to provide a design guide that meets the social and technical design needs identified in chapter 3.

4.2 Introduction to the Collective System Design Methodology

Collective System Design (CSD) is a system design methodology for design of sustainable systems. Developed in the mid 2000's, Dr. David S. Cochran's method comprises the following key components: The Flame Model, the Plan-Do-Check-Act Continuous Improvement Cycle, and the CSD Language. CSD's key components provide a rigorous approach for identifying customers and designing a system to meet customer needs. CSDs separate the means of achievement from the design purpose called the Functional Requirement (FR) [29]. This focus leads to and open framework that facilitates learning and experimentation as the solution is treated as a hypothesis for meeting a FR. CSD has applications in many areas such as hospital redesign, diabetes reversal, course development, and manufacturing [30–32]. The Flame Model, as shown in Figure 4.1 provides a view into the process of how people approach diagnoses, design or redesign systems [33]. The various layers within the flame model are connected and express the different viewpoints within a system.

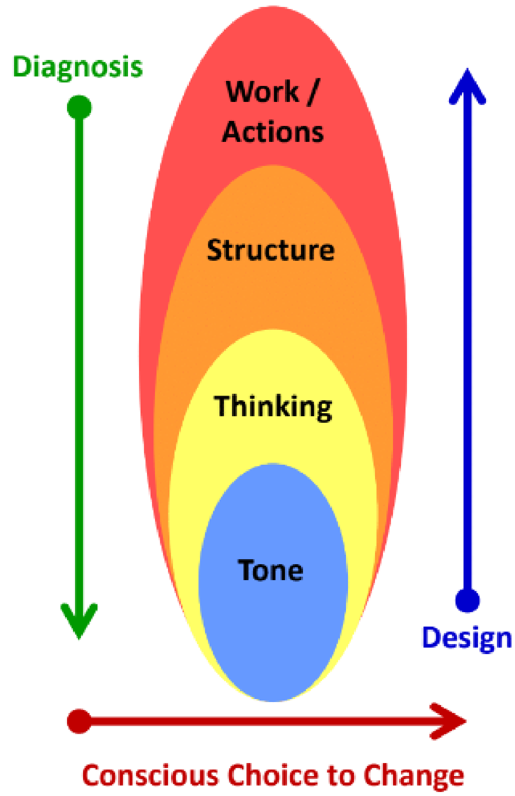

Fig. 4.1. Collective System Design Flame Model. [33]

The hidden driver behind any system design is tone [33]. The tone within system design can be viewed as the design intention and the mental model of what the most important aspects of the design are. CSD acknowledges the importance of tone through its steps, which put emphasis on including the people within the system to establish and express tone in a positive manner [33].

The thinking within the Flame Model is codified through application of Axiomatic Design methodology and derived from the tone/ design intention. The thinking layer translates customer needs into design needs and hypothesizes the means to achieve the design needs. Within CSD, the thinking layer focuses on the customer, functional and design parameter domains and is represented in section 4.2.

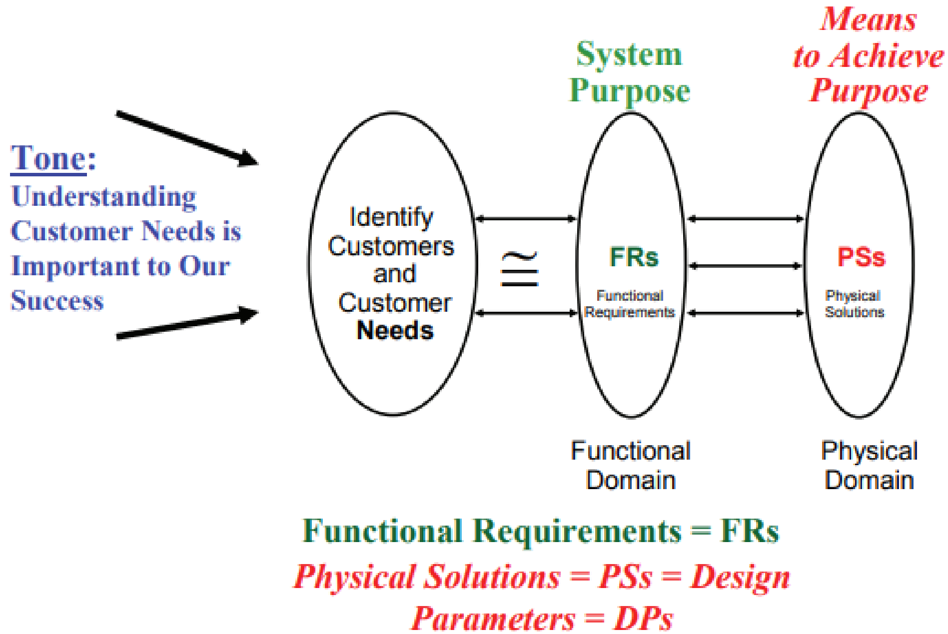

Fig. 4.2. CSD domain mapping translates customer needs to system purpose and means [29]

The sequential order of the domains is important as one domain represents the desired result, and the next domain represents how to achieve it. The process begins by mapping out CNs and their respective constraints. The CNs are used to formulate the top level FRs of the system design. These FRs are mapped into the physical domain by defining one or many proposed solutions or PSs. Each PS typically produces succeeding FRs forming a hierarchy design decomposition. The designer will decompose until implementation of the design can be achieved without further decomposition [30]. Matrix algebra is used to express the independence of each of the proposed FRs. Matrix algebra identifies which FRs are coupled or partially coupled. The critical idea within the thinking layer is to separate the "what" the system must do from the "who" [30]. The goal of separating the system purpose from the means to achieve the system purpose is to choose a PS which only affects the intended FR. The selection of PS then minimizes interactions with other PS's and FRS.

4.2.1 CSD 12 steps

CSD has 12 steps guide the design of a system and in represented in Table 4.1. This thesis is focused on meeting both the designer and customer of product design. While steps 1,8,10,11 are important to system design, these steps will not be discussed they apply more to organizations are outsite the scope of this thesis design.

4.2.2 Decomposition Process and Collective System Design Language

A Collective System Design Map is a hierarchy map which represents the design relationships between customer needs, functional requirements, and physical solutions. An example of such relationships is provided in Figure 4.3 below. The decomposition

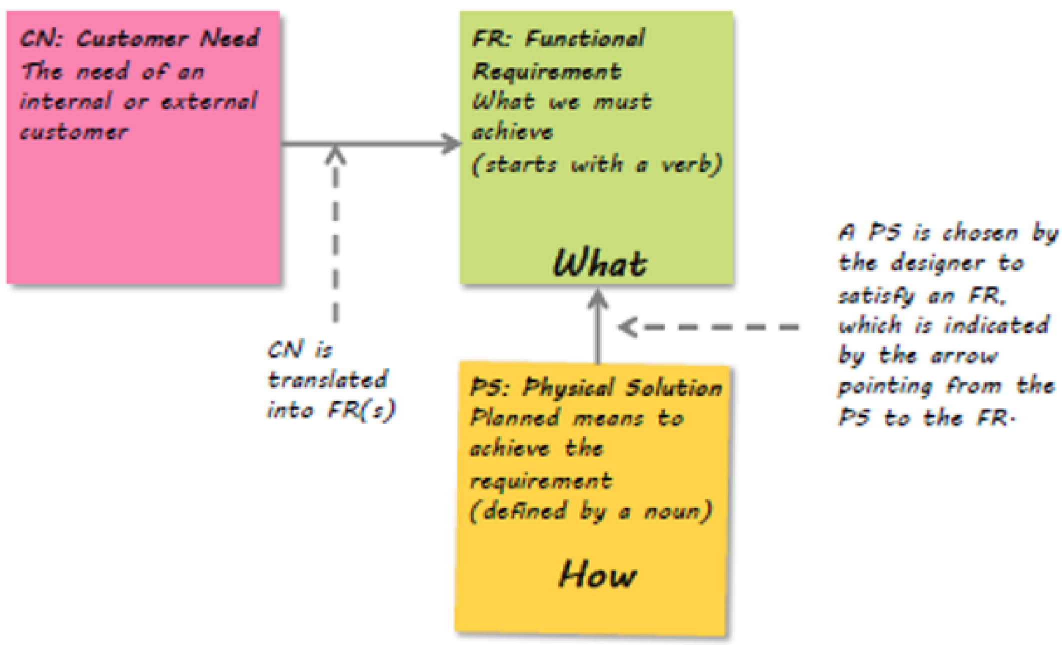

Fig. 4.3. CSD Language enables consistent communication of what the system must achieve and how the system will achieve it

Table 4.1.
Collective System Design 12 Steps [30]

Step	Descriptions	Questions
1	Senior Leadership Makes a Conscious Choice to Change	• Why would we make the change? • Are we capable of achieving something greater? • Is continuous improvement important to us?
2	Define Stakeholders & System Boundary / Value Stream(s)	• Who will be affected by the change? • Who should be involved in the process? • What/who can and cannot be controlled? • What information is passed across the system boundary? • What risks exist within the interfaces?
3	Establish Tone and Values	• What attitude is required to get everyone to participate? • What attitude is required to facilitate collective agreement? • How do we convey to our people the tone we desire for our organization? • How do we problem solve together?
4	Identify Customers and Needs	• Who will purchase/use our product(s)? • What will this customer/user need the product to do?
5	Determine Functional Requirements (FRs)	• How do we state the needs of our customers as functional requirements of the system? • What MUST the system achieve to satisfy the customer(s)?
6	Map the Physical Solutions (PSs) to FRs	• What function MUST the system achieve? • Is the design uncoupled/partially coupled, but not fully coupled? • Are the requirements of the leaf PSs sufficiently clear and implementable through standard work?
7	Define Performance Measures (FR_M & PS_M)	• How will we know if the FR is achieved (FR_M)? • How will we know if the PS is implemented correctly (PS_M)?
8	Define Organization Structure based on CSD Map	• What team structure is needed to make the implementation? • What should the value stream look like? • Can we physically simulate the value stream?
9	Establish Standard Work by Continuous Improvement: Plan, Do, Check, Act (PDCA)	• Currently, what is the best practice for completing the work? • How will we implement a PS as standard work? • Does the current standard work achieve the FRs of the system design? • How can the standard work be improved?
10	Evaluate the Cost of Not Achieving the FRs	• What is the result of not achieving each one of the FRs? • What is the cost benefit of achieving, unachieved FRs?
11	Prepare Resource Re-allocation Plan	• What restructuring of resources is required to make and sustain the system design transformation?
12	Feedback for Sustainability and Growth	• What was the result of the implementation? • What are the required continuous improvement efforts that are necessary to sustain and to improve the system?

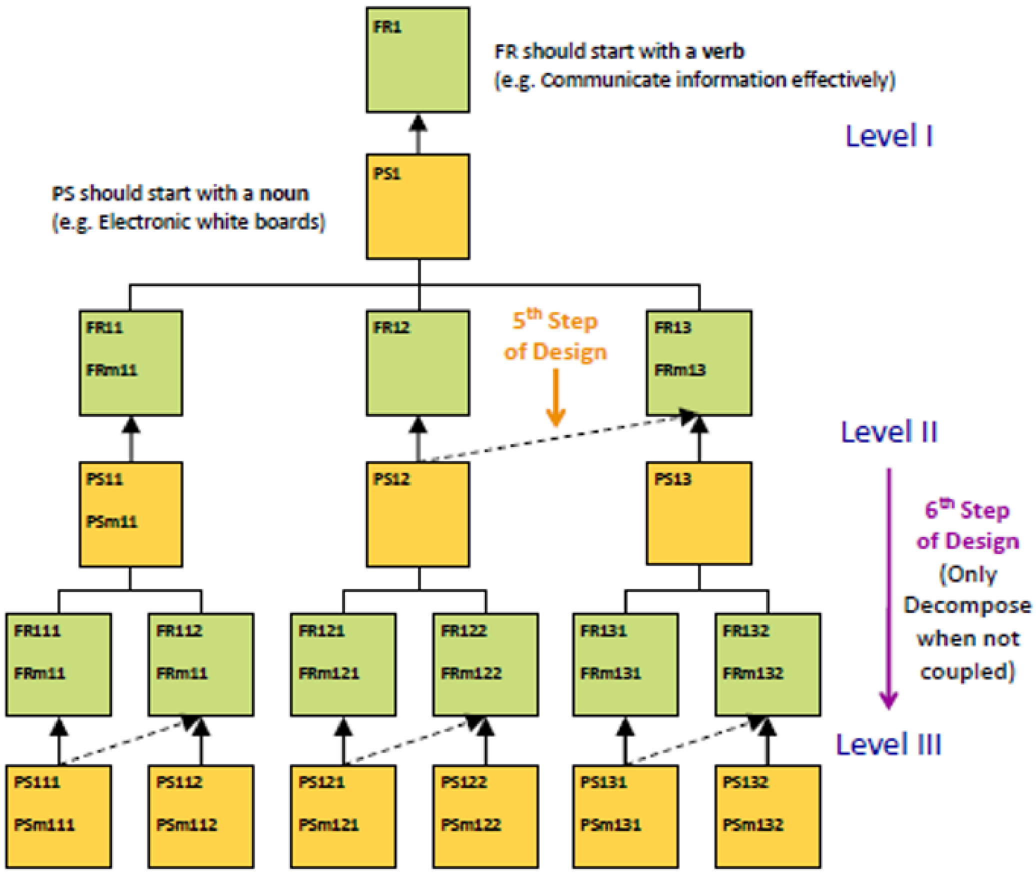

Fig. 4.4. Decomposition Process [30]

process is as follows:

1. Map Customer Needs to Functional Requirements
2. Pick a Physical Solution to meet the Functional Requirement
3. Define Functional Requirement Measures (FRm)
4. Define Physical solution Measures (PSm)
5. Check for interactions or coupling. This is done by asking "does the choice of PS affect the achievement of FR within a specific level and branch".

6. Decompose the design to the next level after performing step 5 and the design is uncoupled or path dependent. Path dependencies are indicated within the decomposition with arrows. The next level can only be decomposed if there are at least two or more FRs from previous PS.

7. Repeat steps three through six until the design is expressed and can be implemented.

The process is illustrated in Fig. 4.4.

4.3 Introduction to Decision Driven Design

Decision Driven Design (DDD) is a systems engineering methodology based on effective decision management [39]. The goal of DDD is to minimize the following common system design faults [34]:

- Poor problem framing during technical planning
- Insufficient traceability from evaluation criteria to higher level requirements
- Poor decision analysis leading to less than best solutions
- Conflicting solutions selected due to decision conflicts and slow communication
- Insufficient implementation process due to solutions not properly blended in the project plan
- Decision faults propagate to next phase due to not capturing patterns and lessons learned

The core elements of DDD strive to improve the quality, speed and execution of any project through minimizing information islands, and imbalances between system engineering disciplines. DDD is comprised of the three core elements: decision-centric informational model, DDD methods engine, and decision patterns [35].

4.3.1 Decision Patterns

A decision pattern is a hierarchical model of the problem domain which helps systems engineers quickly frame projects as a decision network or decision breakdown structure [34]. The solution concept of a product development project can be broken down into common decision patterns. Fig. 4.5 below depicts an example for a general pattern for a product/system development.

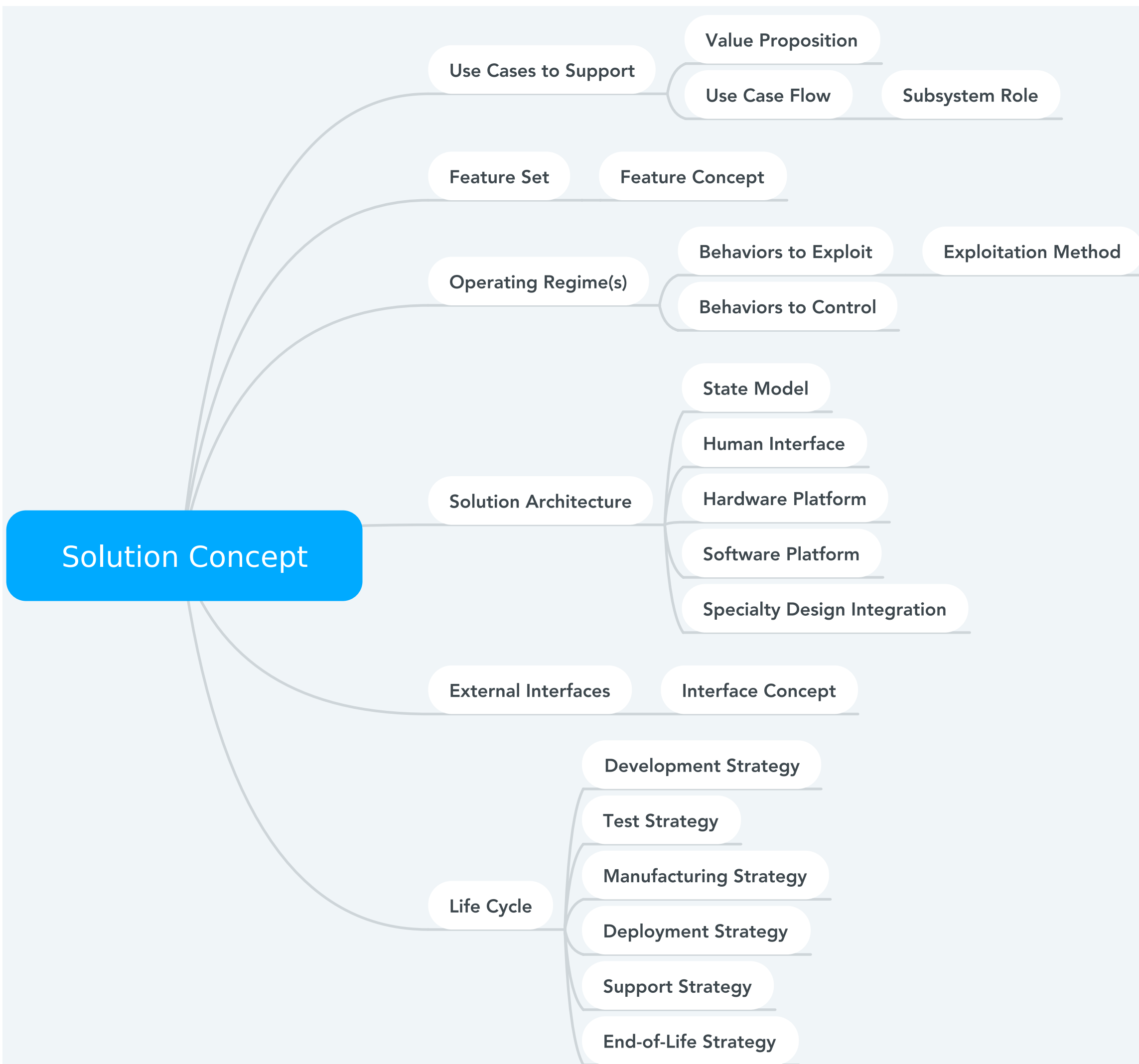

Fig. 4.5. Decision Pattern

The focus of decision patterns is to separate questions/problems from its possible solutions while providing a reusable structure for similar product families or systems [34, 35]. Decision-centric methods allow subject matter experts to preserve their knowledge and influence in designs in the form of a decision pattern.

4.3.2 Decision Driven Design Methods Engine

The DDD methods engine contains three levels of operations. The three operations are plan decisions make decisions and manage decision over time. An overview of the process is depicted in Figure 4.6.

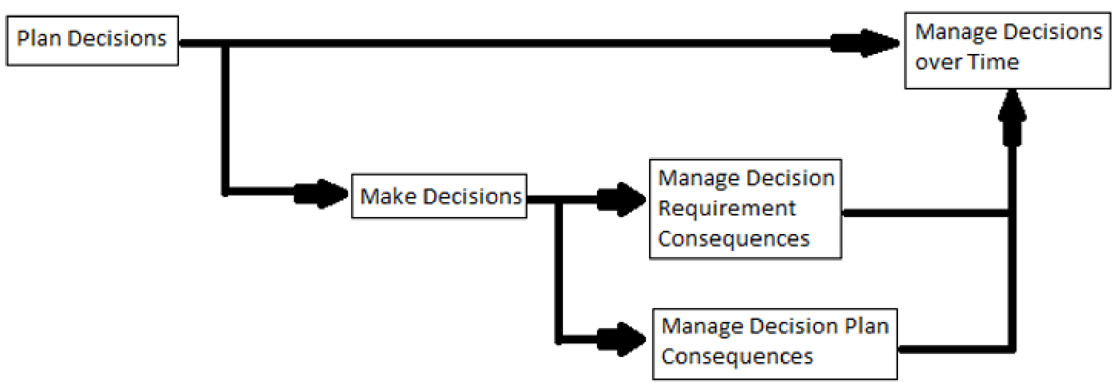

Fig. 4.6. Decision Engine

The consequences of the decision plan and requirements are processed by the designer for each decision made. Evaluation of each decision represents an analysis to select the best course of action per decision and communicates the implications each action has upon the system. The Manage Decisions over Time is a process to anticipate future states and dependencies.

CHAPTER 5. GUIDE FOR CONCEPTUAL DESIGN

5.1 Introduction

This thesis has identified that the design process should provide adequate support through meeting designer needs. The focus of this chapter is to propose a design guide that will meet the social and technical design needs in Table 5.1. The guide will be using concepts from Collective System Design and Decision Driven Design patterns. Table 5.2 below represents the steps to guide the conceptual design processes. The guide will use decision patterns to describe the product development team's design intent and describe the thinking behind each design commitment. The guide will then use the CSD methodology core concepts of tone and design decomposition to map what the system must achieve to how the system will meet each need.

5.1.1 Establish Proper Tone and Values

The Collective System Design starts with tone and defines that the thinking layer within the flame model is biased by the tone. The "Tone" of the product development team will affect every part of the design process. To begin meeting the social needs of the design process, a proper tone for the development team must be set. A proper tone facilitates a respectful, learning environment and enables everyone to engage in creating a system to meet all customer needs. To meet all technical needs, the product development team must practice and maintain proper tone throughout the design process.

Table 5.1.
Social and Technical Design Needs

Social design needs	**Technical design needs**
Desired end state or goal	Capture derivation source of knowledge
Relative significance of product features	Support multiple solution concept analysis
Boundaries to the design problem	Convergence toward a solution
Criteria for evaluating input and output information	Capture needs from multiple sources
Distribution of design responsibilities without loss of system holistic view	Decompose operational and system requirements
Manage uncertainty throughout design	Management of decision consequences
	Assess risk of solutions
	Manage roadmap or plan

Table 5.2.
Steps to guide the conceptual design process

Step	**Description**
1	Establish proper tone and values
2	Define stakeholders and system boundary
3	Identify customers and needs
4	Define research and information collection strategy
5	Determine Functional Requirements (FR)and Functional Requirement Measures (FRm)
6	Map Physical Solutions (PS) to FRs and define Physical Solution Measures (PSm)
7	Define Testing strategy

5.1.2 Define Stakeholders and System Boundary

The thinking layer of the flame seeks to separate what the system must achieve from how the system will achieve its goals. CSD states that before mapping and decomposition can be done, the Stakeholders and system boundary needs must be understood [31]. To enhance CSD, Decision driven pattern is used to capture knowledge sources and to quickly facilitate shared agreement of who the stakeholders are, and to identify the system boundary in table 5.3.

Table 5.3.
The decisions captures the uses cases, and values [34]

Decision number	Name	Description
1	Solution Concept	What is the top-level concept for this system or solution? What makes it unique?
1.1	Use Cases to Support	What use cases (scenarios, missions) will this solution support?
1.1.1	Value Proposition	How will the solution deliver value to the end users and customers of this use case?
1.1.2	Use Case Flow	How will this use case be performed? What flow of activities and events will occur?
1.1.2.1	Subsystem Role (CONOPS)	What role with the subsystem play in this use case? What capabilities and value will it deliver?

5.1.3 Identify Customers and Needs

Through answering each question in table 5.3, The design team has identified the use cases for the product and how the highest subsystem will deliver value. The information in table 5.3 is used to create a hierarchic list of needs. Based on the hierarchy of needs, new groups of customers can be identified. The decision pattern adds a structured approach to framing the design problem and enables the hierarchy of needs to be organized.

5.1.4 Collect Information Strategy

Before needs can be mapped to system Functional Requirements, the maximum amount of information must be obtained. This step is referred to as the Collect Information Strategy. The collect information step is a thorough investigation aimed at collecting mechanical, chemical and electrical information as well as other relevant aspects of the problem, and is captured in Table 5.4.

Table 5.4 illustrates the source of knowledge for how the product will deliver value. Table 5.4 also outlines a research strategy for how the design team will evaluate data.

5.1.5 Determine Functional Requirements

The design team must remember the difference between Physical Solutions and Functional Requirements. Functional Requirements are the system's answer to what the system must achieve to meet the needs of the customer. Physical Solutions are the system designer's tools inherent in the product used to achieve the Functional Requirement. Ideally the Functional Requirement is defined before the Physical Solution. There are times in product development where a few Physical Solutions are predetermined by the customer and must be used in the product. One example is where a product is redesigned and the previous hardware platform must be reused due to cost constraints.

Table 5.4.
Decisions describe how the design team will investigate and gather the necessary information for the project

Decision number	Name	Description
1.3	Operating Regime(s)	In what range of conditions, environments and performance levels will the solution operate?
1.3.1	Research Strategy	What research strategy will we use to understand the science associated with this solution's operating regime? What set of studies and experiments will we conduct? How will these experiments interact to create a body of knowledge?
1.3.2	Behaviors to Leverage	What behaviors or properties (science) within the operating regime will be leveraged to create value?
1.3.2.1	Leverage Method	How will the solution leverage this behavior/property to deliver value?
1.3.3	Behaviors to Control	What behaviors or properties (science) within the operating regime will be controlled (regulated, suppressed or avoided) to realize value?
1.3.3.1	Control Method	How will the solution control or suppress this unwanted behavior or property?

Features are considered the prominent selling characteristics of the product. An example for a feature would be low cost modular design. The features are captured by decisions in the table 5.5.

The design problem can now be expressed in a more concise approach. The new specification of the problem must be clear, concise and general. The goal is to restate the problem in terms of the list of needs and features mapped to a top-level Functional

Table 5.5.
Decisions describe the features of the product

Decision number	Name	Description
1.2	Feature Set	What are the primary features or groups of features that will be delivered?
1.2.1	Feature Concept	How will this feature be implemented (technology, top-level design)?

Requirement model. The decisions to create the restatement of the design problem are represented in table 6.4.

Table 5.6.
Decisions describe the top level requirements and solution architecture

Decision number	Name	Description
1.4	Solution Architecture	What is the solution's top-level architecture; the allocation of functions to hardware, software or user actions? What level of automation will be provided (automation boundary)?
1.4.1	State Model	Which life cycle and operational states will the solution support? What is its top-level state-machine representation?
1.4.1.1	Modes	Which modes will the solution support in this state?
1.4.2	Functional Requirement Model	What is the solution's top-level functional model/ architecture (functional flow, relationships)?

The Functional Requirements (FR) model is the smallest set of independent requirements that serve to describe the design objectives [30]. The author states that each FR must be constrained and the requirements for when the FR is achieved must be defined. The measures on a FR are referred to as Functional Requirement measures (FRm) [31]. For example: to accelerate a vehicle could be an FR, and a range of 0 to 60mph in 4 seconds would be a FRm. An FRm is one of the two types of constraints. Constraints limit the acceptable design solutions. The two types of constraints are input and system constraints. Input constraints are design specifications, such as cost and physical limits. System constraints or FRms are constraints imposed by the system for which solutions must operate. Decision criteria is used to define FRms and refine FRs. Table 5.7 represents a decision about technology. The criteria within that technology decision should support the design team in describing additional FRms.

5.1.6 Map the Physical Solutions (PSs) to FRs and define Physical Solution Measures (PSm)

In answering the questions in Tables 5.3 - 6.4, the design process has met the following needs:

- Defined the end state or goal
- Bound the design problem
- Capture derivation source of knowledge
- Capture needs from multiple sources
- Criteria for evaluating input and output information

Meeting the above needs enables the design team to move to the next step of mapping the PSs to FRs and defining Physical Solution Measures. PSm captures whether the solution is implemented [30]. The following decisions in Table 5.8 describe

Table 5.7.
Decisions criteria is used to refine FRs and define additional FRms

Decision number	Name	Description
1.4.2.1.1	Technology	What technology, method, design approach or algorithm will be used to deliver this function?
Criteria	Safe	The technology should be safe in all phases of the product life cycle
Criteria	Performance goal 1	The technology should perform well against this goal or parameter
Criteria	Performance goal 2	The technology should perform well against this goal or parameter
Criteria	Simple	The technology should be simple; easy to set up, control and use
Criteria	Output compatibility	The technology should deliver the outputs expected for this function
Criteria	Input compatibility	The technology should match inputs for this function
Criteria	Reliable	The technology should be reliable; fail seldom and fail softly
Criteria	Robust	The technology should be robust; produce repeatable results despite input variations and operating conditions
Criteria	Footprint	The technology should have a small footprint (volume, mass, power, waste, etc.)
Criteria	Low risk - proven	The technology should have a proven track record of success; have a low risk of technical failure
Criteria	Low cost	The technology should have low development, production and operational costs

and assess the conceptual base and the development team's capability of implementing each proposed solution. Some decisions will have multiple solutions proposed as a decision could encompasses multiple FRs and impact multiple PSs.

From the information and alternatives gathered in Table 5.8, the PSs can be mapped to the FRs and the CSD decomposition process can begin. This is illustrated below in figure 5.1

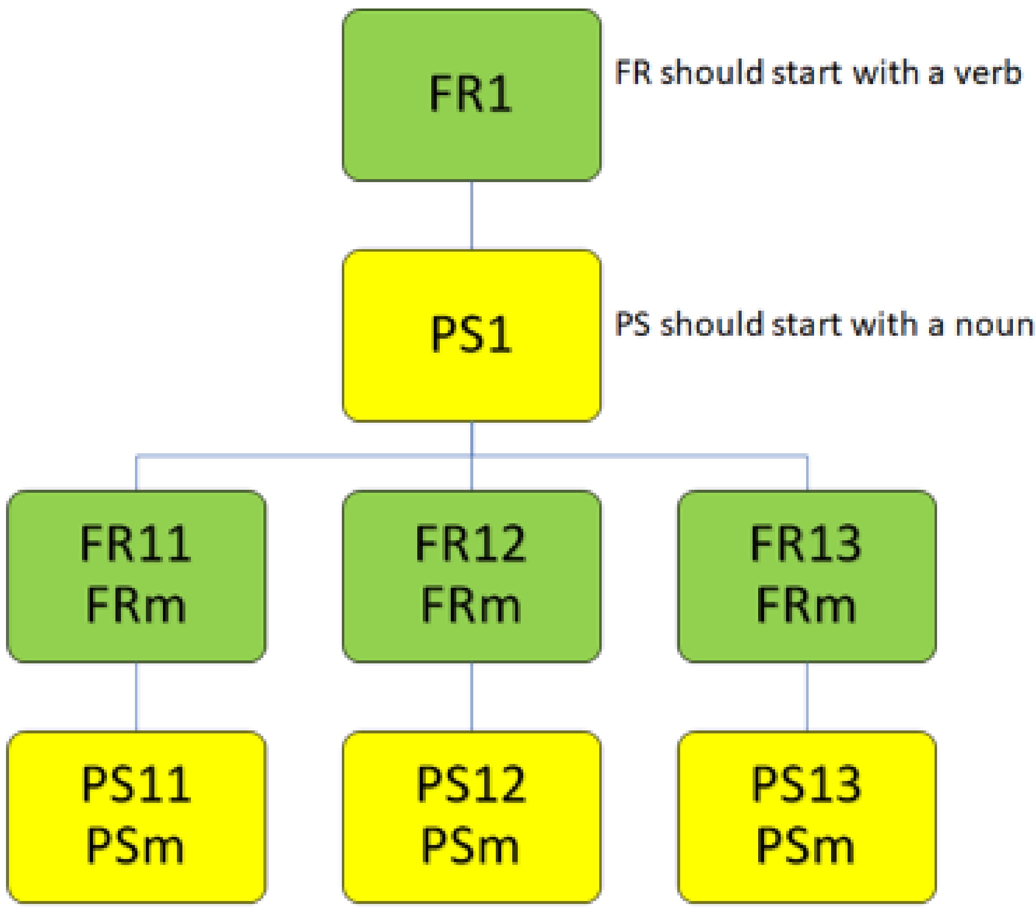

Fig. 5.1. Mapping PSs to FRs

Table 5.8.
Decisions describe the capabilities of the development and access the conceptual base of each alternative proposed

Decision number	Name	Description
1.4.2.1.1	Technology	What technology, method, design approach or algorithm will be used to deliver this function?
1.4.2.1.2	Functional Interfaces	What function-to-function interactions will the solution support?
1.4.2.1.2.1	Interface Concept	How will this functional interface be implemented?
1.4.3	Human Interface	What type of human interface will the solution provide?
1.4.3.1	User Tasks	What user actions/ tasks will the solution support?
1.4.3.1.1	Interaction Method	How will the user interact with the solution to accomplish this task?
1.4.3.2	Data Presentation	How will the solution present information (data) to the user?
1.4.3.3	Control Presentation	How will the solution present control information (e.g. menus) to the user?
1.4.4	Hardware Platform	What hardware platform will deliver/ host this solution?
1.4.4.1	Hardware Standards	Which hardware standards will we follow for this solution?
1.4.4.1.1	Compliance Strategy	How will our solution be designed to comply with this standard?
1.4.4.2	Form Factor	What form factor (mechanical packaging concept) will be used for the solution?
1.4.4.3	Hardware Architecture	What is the solution's hardware architecture (hardware elements, allocated functions)?

Once the design team has the information to satisfy a particular requirement, they apply the following Axioms.

1. Maintain the independence of the functional requirements

2. Minimize information content

The first Axiom dictates that solutions should not break the independence of the functional requirements. The first axiom limits the number of acceptable alternatives or solutions that are available. The second axiom establishes that the best designs and solutions are those requiring the lowest information content to satisfy the FRs. The information content is the uncertainty associated with the probability of success [36]. Therefore, alternatives with the lowest information content should be selected when mapping to PSs. PSs and design components should be traceable to testing and evaluation procedures. When testing before moving to the next level of the decomposition, the design team assess the path dependency of each PS. The path dependency is assessed by asking the following question: Does the choice of a PS affect the achievement of each FR in the same branch? If the answer to this question is yes, then the design team would place an arrow from the PS to each of the affected FRs. Figure 5.2 illustrates this process.

In figure 5.2 PS11 will affect the achievement of FR11, FR12, and FR13 and is referred to as partially coupled. Partially coupled and uncoupled are acceptable designs because they have predictable system outputs. Uncoupled is where one PS will satisfy one FR. A coupled design is illustrated in figure 5.3 below and is indicated by the red arrow. CSD dictates that coupled designs are unacceptable and must be corrected before decomposing the system to lower levels [30].

The decomposition continues until the design can be implemented by the design team. The completed decomposition represents the road map for implementation of the design.

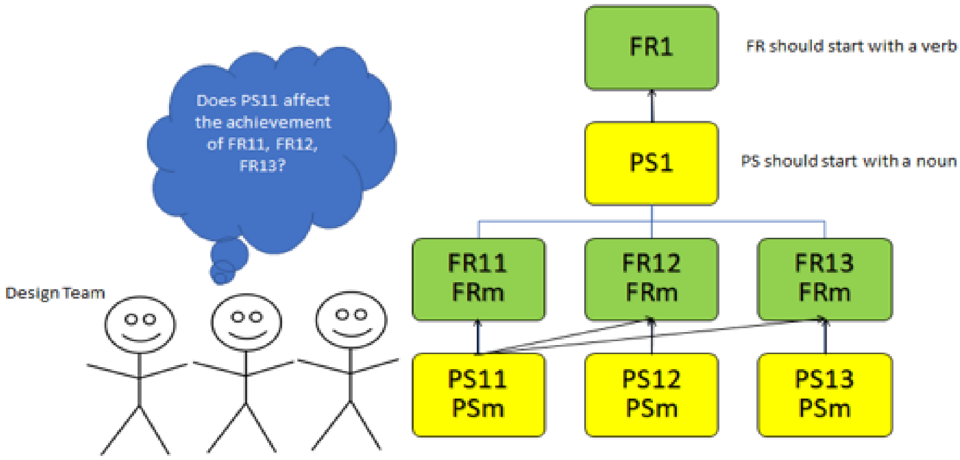

Fig. 5.2. The design team assesses the path dependency of each proposed solution or PS

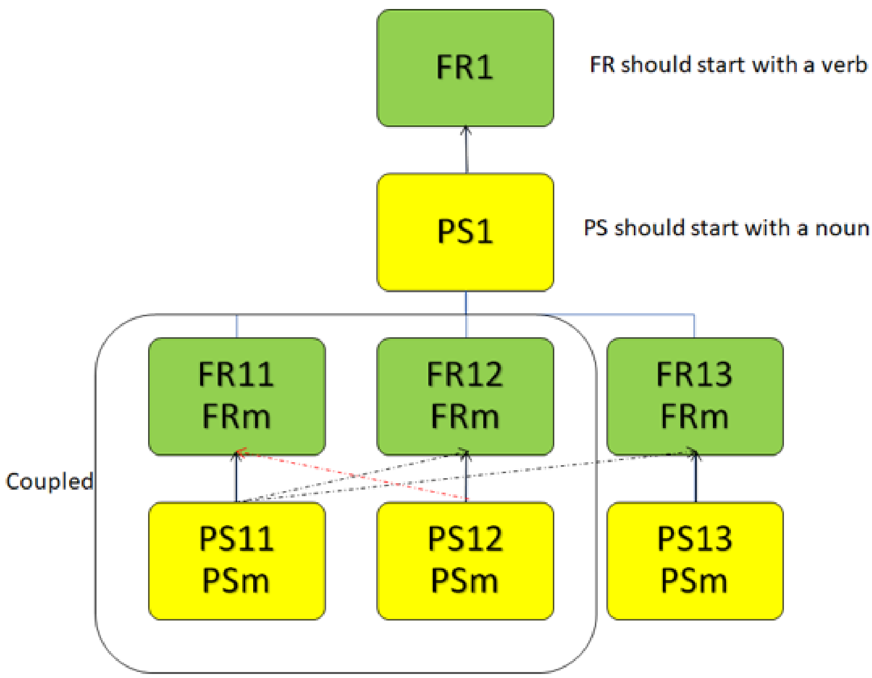

Fig. 5.3. The coupled design is indicated by the red arrow from PS12.

5.1.7 Test Strategy

A test strategy will look different for each team, but must define and focus on test objectives to be successful. The following decisions in table 5.9 lay the foundation for the testing strategy and its objectives. A design failure mode and effects analysis (DFMEA) should follow the decomposition process to identify additional FR-PS pairs.

Table 5.9.
Decisions describe testing method for the project

Decision number	Name	Description
1.6.2	Test Strategy	What strategy will we use to test this solution?
1.6.2.1	Test Methods	What test methods will we use to verify this solution meets its requirements?
1.6.2.2	Test Tool Suite	What suite of test tools will we use to verify this solution?
1.6.2.3	Test Team	Who will staff the testing effort? What role will each play?
1.6.3	Trial Strategy	What strategy will we use to trial this solution before it is released to market? What set of trials will we conduct? How will these trials interact?
1.6.3.1	Trial Concept	What is the top-level design concept for this trial? (scope, participants, methodology, outcomes)

There will be cases in design when the PS will not satisfy the intended FR. A Plan Do Check Act (PDCA) from CSD will be used to improve the design of the product. The Plan of the PDCA describes the details of how to implement the proposed PS. The Do step is where the product development team implements the plan. The Check step uses predefined standard work to test if the PS was implemented and that the PS has met the intended FR. The Act step determines if the design should be changed

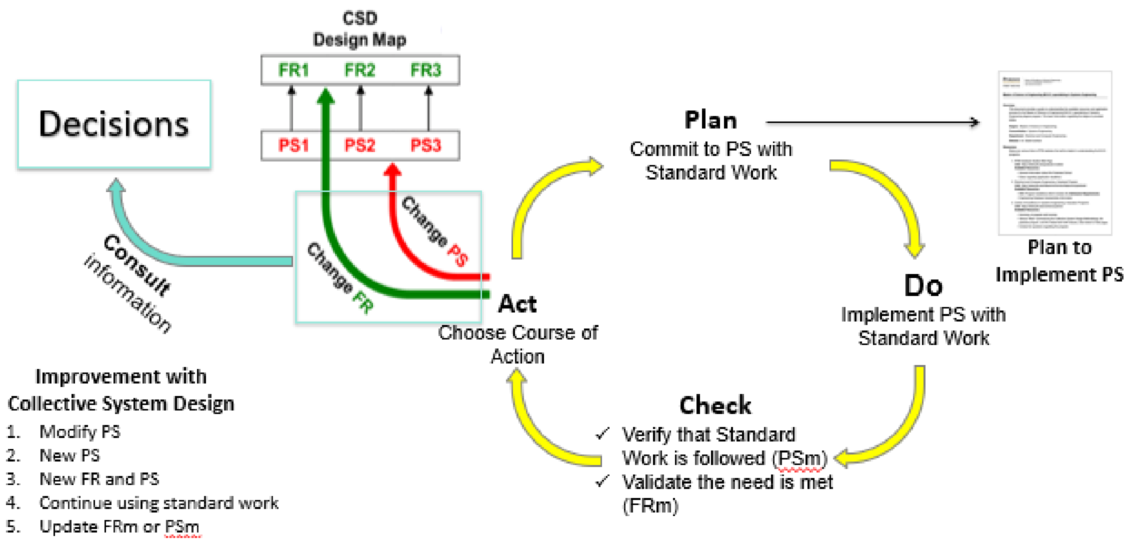

Fig. 5.4. The PDCA is a continuous improvement process that enables the design to team to make product design changes

based upon the testing results. The author augments the CSD PDCA by adding in a decision component. The author assert that if a change to an FR or PS is needed, the designer must consult the decisions and decision criteria which lead to the original FR definition and PS commitment. Consulting the information described within the decisions will enable the design team to stay aligned with the original design intent and constraints. Figure 6.12 illustrates this process.

5.2 Summary

The Collective System Design (CSD) methodology along with the Decision Driven Design (DDD) have met the social and technical design needs identified in Table 5.1. The tone is represented through the shared commitment during the design decisions and CSD decomposition processes. Shared agreement during each process ensures that every team member knows what the customer needs are and how to implement a quality solution. The decision pattern describes the information source encompassed by the Functional Requirements and how such information impacts the Physical Solutions. The CSD decomposition overall provides a road map of implementation and design structure. Chapter 6 follows the proposed design process with a design team at Purdue Fort Wayne.

CHAPTER 6. JILL PROSTHETIC HAND DESIGN

6.1 Introduction

Chapter 5 the covered the integration of Collective System Design (CSD) and Decision Driven Design (DDD). The integration of the two design methods met both the social and technical needs identified in Chapter 3. This chapter will cover the last research objective and apply the integrated methods to the conceptual design of a product. This research objective will show the integrated method can describe and design an innovative product while maintaining a holistic system view and traceable elements. The objective is to produce an affordable, comfortable, and easy to maintain prosthetic hand for trans-radial and wrist disarticulation patients. The conceptual design of the product will use the integrated CSD and DDD method. The design of the Jill prosthetic system requires the patient to have had a functional hand. This design assumes the patient has the necessary functional muscle groups for the sensor actuation of the prosthetic.

6.2 Establish Tone and Values

The design team's approach to the design of the Jill prosthetic system provides more than just the prosthetic device. The team aims to create a prosthetic hand system that will improve the lives of trans-radial amputee patients. The Jill prosthetic system is intended to be available to people regardless of financial background. The Jill prosthetic system will help with mental trauma and facilitate therapy to enable patients to move forward in life.

6.3 Define Stakeholders, System Boundary and Identify Needs

The key idea of this design step is to take decisions from decisions patterns and establish which use cases the top level solution will support. Step 2 is very crucial to product design as it is where most of the system needs will be derived from the use cases. The use cases describe how the customer would want to use the Jill prosthetic system. The customer needs are generated from the use cases in Table 6.1. The needs are very important as they provide the foundation for the Jill prosthetic system.

6.3.1 System Boundary

The system boundary shown in Figure 6.1 defines the enterprise entities within the system and those within the environment. The entities within the system boundary are able to be controlled as the Jill prosthetic is developed, produced, tested and marketed. The stakeholders and customers include: Product development teams, rehabilitation teams, call center teams and traveling technician teams. The internal entities interface with others outside of the Jill prosthetic enterprise such as doctors, charitable foundations willing to pay for the prosthesis, sales and marketing. Table 6.2 describes the interactions between each customer in figure 6.1.

6.4 Collect Information Strategy

The collect information strategy captured in table 6.3 is a thorough investigation aimed at collecting mechanical, chemical and electrical information as well as other relevant aspects of the problem. The Jill prosthetic will offer a multitude of grip and gesture options. The development team will need to incorporate mechatronics, kinematics, kinesthetics and kinesiology research to produce a product that will mimic human hand motion. Application of 3D printing technology and off-the-shelf microprocessors enable the end product to be customized for the patient at a very low cost.

Table 6.1.
Displays the used cases to needs

Use Case	**Need**
Patient wants to pick up objects of cylindrical shape	Broom grip
Patient wants to pick up objects like keys	Key grip
Patient wants to pick up round ball like objects	Ball grip
Patient wants to pick up paper-thin objects	Paper grip
Patient wants to pick up cutlery	Cutlery grip
Patient wants to pick up squirt bottles	Squirt bottle grip
Patient wants to hold mouse	Mouse grip
Patient wants the ability to point with desired fingers	Pointing gesture
Patient wants to form a peace sign	Peace Sign
Patient wants to wave to another person	Waving
Patient wants to perform all functions in all weather	Prevent environmental damage to product
Patient wants to low cost in maintenance	Modular design
Jill prosthetic must reach wide range of customers	Accommodate variation in patient needs
Provide patient with multiple ways to fund prosthetic	Provide various options to fund prosthesis
Provide option for direct payment	Direct payment option
Patient can not afford to pay for prosthesis	Financial Aid
Patient has mental trauma from loss of hand	Mentally prepare user for life with prosthetic
People recognize our brand and services	Brand recognition
Patient wants face to face support	Facilitate customer to customer interaction
Patient has right product and knows how to use it	Identify customer and needs
Patient has trouble using the product	information feedback system
Patient communicates with developers	Direct simple communication
Patient has typical questions	Access to FAQ information
Patient's product critically fails	Service the product
Patient struggles with product operation or has questions not answered in the FAQ	Rapid response from operation specialist

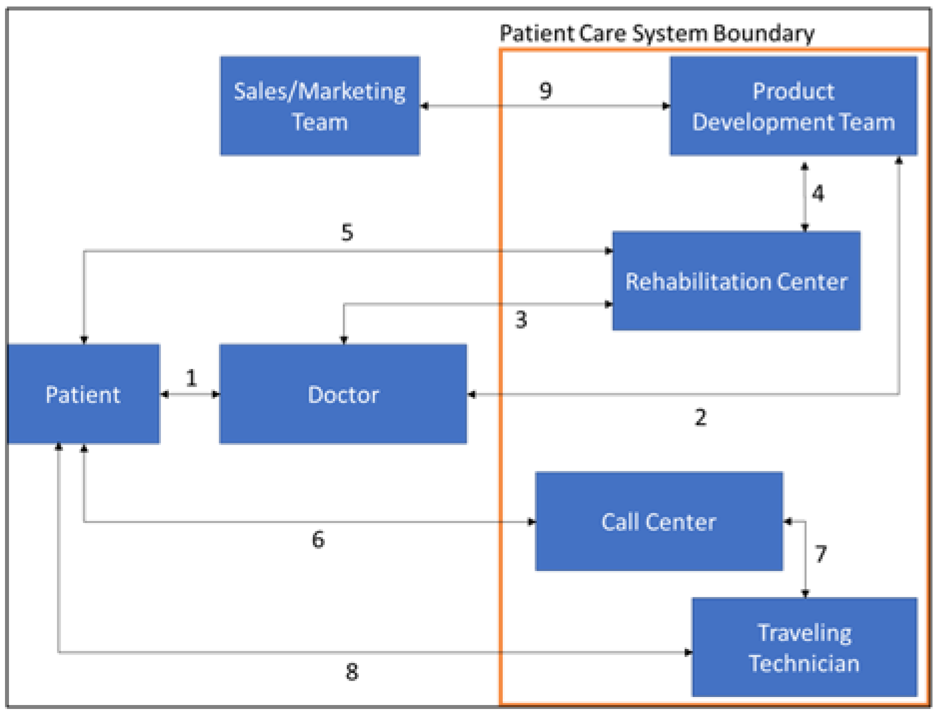

Fig. 6.1. Jill prosthetic system boundary

Table 6.2.
System boundary interactions

Number	Interaction
1	The patients' initial measurements are taken by a doctor
2	The doctor sends the initial assessment results to product development team to initiate prosthesis development
3	The doctor refers the patient to Jill rehabilitation center to finish the assessment
4	The rehabilitation center sends the complete assessment results to the product development team to complete the prosthesis fitting process
5	The patient attends rehab classes and receives product operation training
6	The patient calls the call center for operation issues with the product
7	The call center contacts the traveling technician concerning any prosthetic failures that need to be addressed
8	The traveling technician goes to the patient and repairs or maintains the prosthetic onsite
9	The sales and marketing communicate with the product development team to ensure the Jill prosthetic system to ensure payment

Table 6.3.
Collect information strategy

Decision	1.3	Operating Regime(s)	In what range of conditions, environments and performance levels will the solution operate?	Under normal room temperatures and pressure conditions. Water resistant(IP 24 compliant) and fine dust resistant
Decision	1.3.1	Research Strategy	What research strategy will we use to understand the science associated with this prosthetic device	Mechatronics, kinematics, Kinesiology, Kinesthetics and mature technology usage and research
Decision	1.3.2.2	Leverage Method	How will the solution leverage this behavior/property to deliver value?	Lowcost Microcontroller
Decision	1.3.3	Behaviors to Control	What behaviors/ properties (science) within the operating regime will be controlled (regulated, suppressed or avoided) to realize value?	Multiple grips, multiple gestures, control power, user confusion
Decision	1.3.3.1	Control Method	How will the solution mitigate user confusion?	User mode selection, Proper training in usage

6.5 Determine Functional Requirements

The Jill prosthetic design space can now be expressed in a more concise manner. In this step, the Jill prosthetic is expressed in terms of the list of needs and is mapped to a top-level Functional Requirement model. Each functional requirement is tied back to a customer need to establish requirement traceability. The mapping of customer needs to Functional Requirements (FR) found in Table 6.5. These FRs drive the design of the Jill prosthetic and describe the requirements of the product in order to meet customer needs.

6.6 Map the Physical Solutions (PSs) to FRS and Define Physical Solution Measures

Table 6.6 illustrates how design decisions create a knowledge pull and assess the conceptual base of the development team's capability of implementing each proposed solution. The alternatives are the proposed solutions to each decision. Some decisions will have multiple alternatives proposed as a decision could encompasses multiple FRs and impact multiple PSs. The decisions were connected to each PS to add traceability and describe how each PS was chosen. This traceability proved to be very valuable when selecting the PS that would satisfy the motor-actuation FR. A technology decision and its criteria is detailed in Table 6.5. The Raspberry Pi was chosen over the Arduino development board due to its superior processing power and because it uses a programming language that most of the product development knew very well.

6.6.1 Design Decomposition

Design of the Jill prosthetic is driven by use cases, needs and FRs. The Design Decomposition is developed from these inputs and depicts the relationship between Functional Requirements (what the system must achieve) and Physical Solutions

Table 6.4.

Needs mapped to FRs

Needs	Functional Requirement
Broom grip	1.1.1.2 Stable Grip
Key grip	1.1.1.2 Stable Grip
Ball grip	1.1.1.2 Stable Grip
Paper grip	1.1.1.2 Stable Grip
Cutlery grip	1.1.1.2 Stable Grip
Squirt bottle grip	1.1.1.2 Stable Grip
Mouse grip	1.1.1.2 Stable Grip
Pointing gesture	1.1.1.3 Communicate with Prosthesis
High Five	1.1.1.3 Communicate with Prosthesis
peace sign	1.1.1.3 Communicate with Prosthesis
waving	1.1.1.3 Communicate with Prosthesis
Prevent environmental damage to product	1.1.1.1.1 Comply with IP24
Modular design	1.1.1.1.2 Reduce Cost in General Maintenance
Accommodate variation in patient needs	1.2 Accommodate Variation in Patient Needs
Provide various options to fund prosthesis	1.2.1 Fund Prosthesis
Cheap	1.2.1.1 Provide Direct Payment Options
Financial Aid	1.2.1.2 Provide options fro Financial Aid
Mentally prepare user for life without their hand	1.2.2 Accomodate Mental health Needs Due to Trauma
Brand recognition	1.2.2.1 Reach Customers Who Need Our Product
Facilitate customer interaction	1.2.2.2 Facilitate Customer to Customer Therapy
Identify customer and needs	1.2.3 Accommodate Variation in Technical Needs
information feedback system	1.2.4 Gather Product Feedback from Patient
Direct simple communication	1.2.4.1 Provide Product Operation Aftercare Help
access to FAQ information	1.2.4.2 Answer Frequently Asked Questions
Service the product	1.2.4.3 Repair Critical Product Failures
Rapid response from operation specialist	1.2.4.4 Facilitate Fast Response to Operation Questions

Table 6.5.
Technology decision to meet the FR actuate motors

	Number	Name	Description	Alternatives	
Decision	1.4.2.1.4	Technology - Motor Actuation and processing	What microcontroller or psoc will be used to actuate motors?	Raspberry pi	Ardruino
Criterion		Low risk - proven	The microcontroller must be mature technology with good track record	The raspberry pi has a proven track record of success; has a low risk of technical failure	The Arduino is very mature and lots of documentation for various functions
Criterion		Low cost	The microcontroller must be low development, production and operational costs. Overall under 70	Cost is 49.99	Cost is 36.99
Criterion		Rapid implementation	The microcontroller must be able to be implemented quickly	The raspberry pi is plug in play and includes an up to date data sheet	Arduino is plug in and play and includes up to date documentation
Criterion		Fit our capabilities	Does the microcontroller match our core competencies and resources	Uses python programing which most of our team knows	Uses proprietary program language similar to C++
Criterion		Output sufficient power	The microcontroller power must output sufficient to power and actuate all 4 motors at 3.7V 1A	The Raspberry pi has 3v and 5v output with up to 2A	Output limits at 5V and 1A
Criterion		Process EEG signals	The Microcontroller must be able to process EEG signals	The raspberry pi has multichannel process capabilities	The Arduino interfaces with hardware easily and but has multiple issues with large sofware applications

Table 6.6.
Technology decisions and functional interfaces

Decision	1.4.2.1.1	Technology	What technology or method will be used to sences the EEG signals from the muscles	Myoelectrical sensor
Decision	1.4.2.1.3	Technology	What technology will be used to provide the necessary torque to move the prosthetic fingers	4 dc brushed motors - uxcell brand 300 rpm
Decision	1.4.2.1.4	Technology	What microcontroller will be used to actuate motors?	Raspberry pi
Decision	1.4.2.1.2	Functional Interfaces	What function-to-function interactions will the solution support?	Control motors, actuate motors, Recieve muscle EEG signals
Decision	1.4.2.1.2.1	Interface Concept	How will this functional interface be implemented?	Proper sensor placement
Decision	1.4.2.1.2.1	Interface Concept	How will this functional interface be implemented?	Preprogrammed control labels
Decision	1.4.2.2	Technology	What technology, method, design approach or algorithm will be used to deliver preprogrammed control labels	Python programming

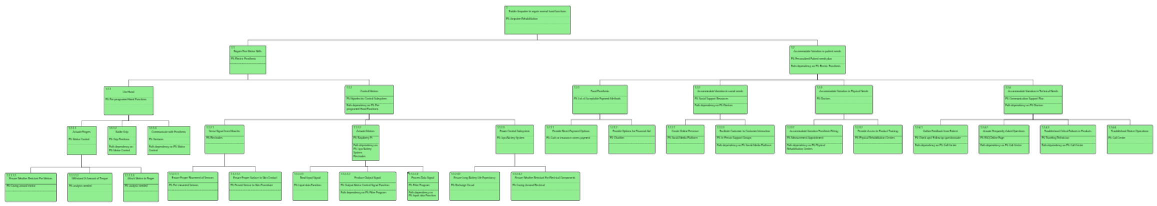

Fig. 6.2. Jill prosthetic decomposition created with Genesys requirement software

(how the FR will be met). The top FR requires amputee patients to regain normal function; this FR is satisfied by a PS1: Amputee Rehabilitation. The decomposition is captured in a requirements management program called Genesys [37]. Genesys enabled the development team to map PSs and to assess path dependency. The partially coupled branches are indicated by the path dependency text below a PS. The full decomposition is shown in figure 6.2.

Figure 6.3 provides an overview of the prosthetic system boundary and identifies the mechanical and electrical interfaces. Table 6.7 defines the interfaces between major components within the Jill prosthetic.

The top-level functional requirement is to enable the amputee to regain normal hand function. The proposed physical solution is amputee rehabilitation. The rehabilitation of the patient requires our system to enable the patient to regain motor control and for our system to accommodate variation in each patient's needs. Figure 6.4 depicts the top-level decomposition of the Jill prosthetic system. Each branch is described in following sections.

6.6.2 Regain Motor Control Branch

The focus of the regain motor control branch was to design a subsystem to enable the patient to perform essential grips and gestures. The patient will be controlling the motors using a myoelectric control system. Myoelectric control is where the patient

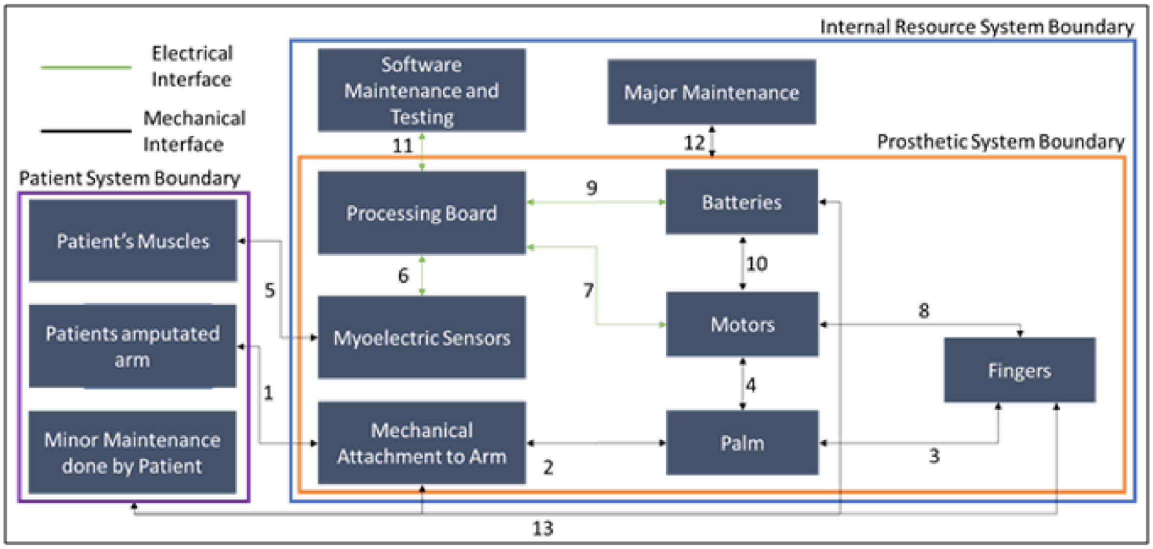

Fig. 6.3. Jill prosthetic product system boundary

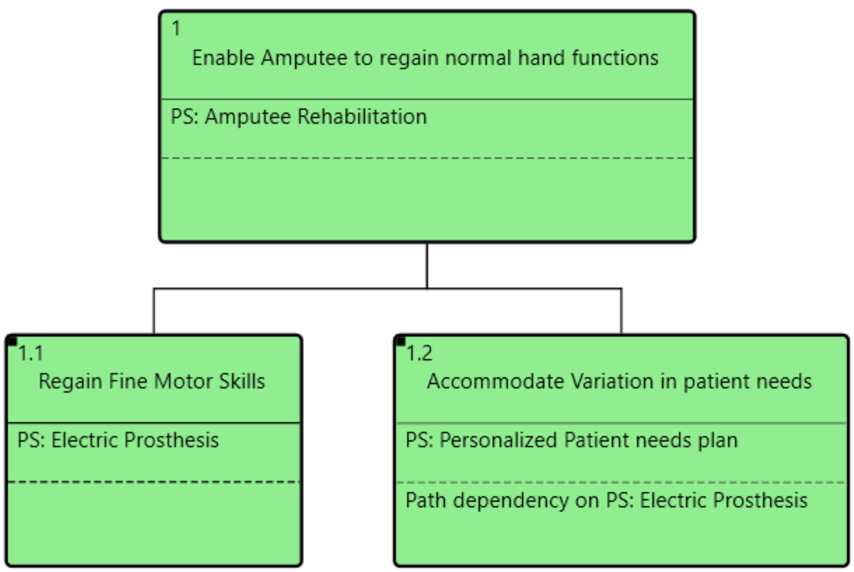

Fig. 6.4. Top level decomposition

will flex a target muscle group to generate an electrical signal that will control the functions of the Jill prosthetic hand. One risk is patients may not use the prosthetic

Table 6.7.
Prosthetic product interfaces and system boundary

Number	Interface Description
1	The patients arm is fitted within a sleeve and secured
2	The sleeve is connected to the palm using standard connector
3	The palm is mechanically connected to each finger through a hinge and string system
4	The palm provides mounts for motors placement
5	The myoelectric sensors are placed on the patient's muscles
6	The myoelectric sensors transmit EEG signals to the processing board
7	The processing board actuates the desired motors
8	The motors move fingers through a hinge and string mechanism
9	The battery powers the processing board
10	The battery powers the motors
11	Software maintenance test the programs and functions of the prosthetic device
12	Maintenance may need to be performed on the prosthetic system
13	The patient may perform minor system maintenance

due to difficult prosthetic operation. The PS preprogrammed hand functions will enable the patient to use the hand with ease in any predefined way. The decomposition is shown in Figures 6.5, 6.6, 6.7.

In order for the patient to use the hand's preprogrammed functions, the program must actuate the fingers, provide stable grip, and communicate with gestures. The decomposition below the preprogrammed hand functions is illustrated in Figure 6.6. The PS motor control will affect the achievement of the FRs stable grip and communicate with prosthesis. This is indicated by the path dependency. The decomposition below the PS motor control is shown in Figure 6.7. To have sufficient motor control FRs, the motors are required to withstand X amount of torque. The FRs states X amount torque due to the fact that more analysis is needed to define the interface between the motor and the control mechanisms. The FR ensure weather resistance

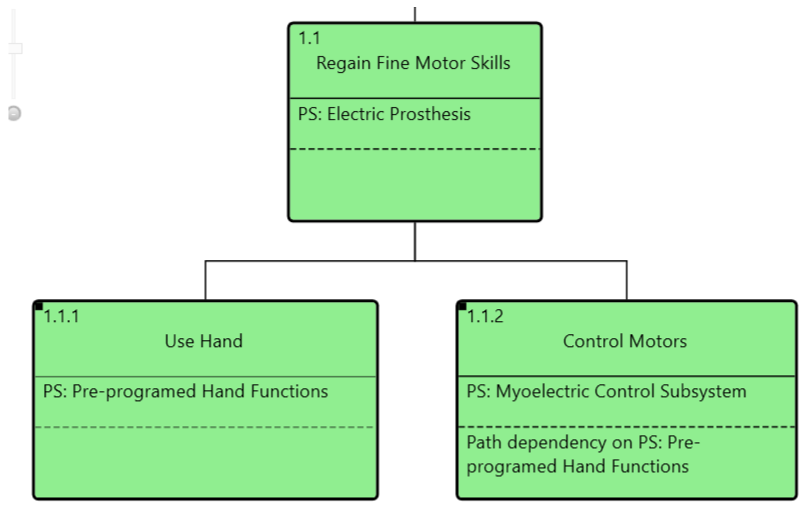

Fig. 6.5. Regain motor control branch

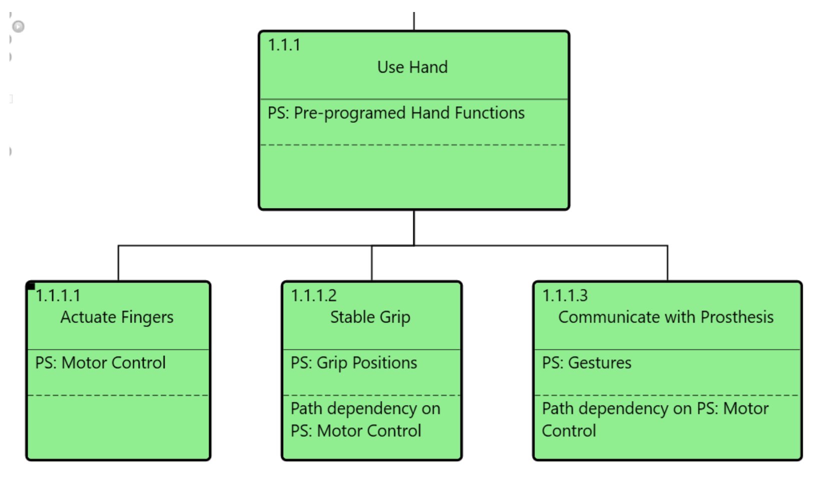

Fig. 6.6. Decomposition of the PS: pre-programmed hand functions

for motors means there must be some technology or material in place to protect the motors from the environment. The PS chosen was to have the motors encased to protect from environmental damage.

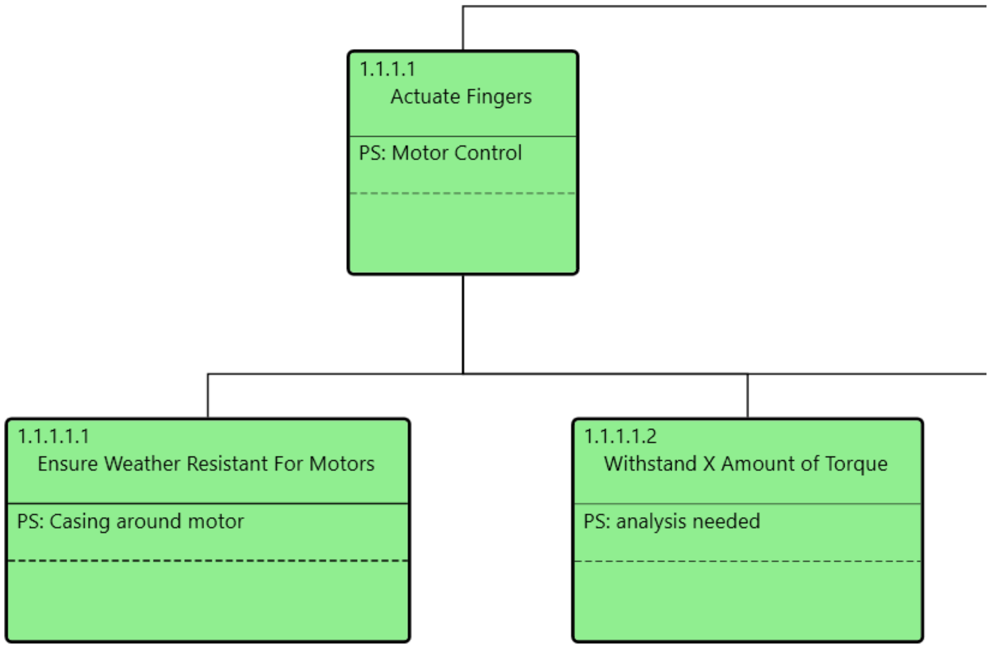

Fig. 6.7. Decomposition of motor control system

6.6.3 Accommodate Variation in Patient Needs

The focus of the accommodate variation in the patient needs branch was to design a subsystem to meet not just physical needs, but mental and technical needs as well. The Jill prosthetic does more than design a prosthetic but also provides additional mental support.

The decomposition below the PS list of acceptable payment methods is illustrated in Figure 6.9 and requires the Jill prosthetic system to provide direct payment and financial aid options. The mission is to improve the lives of transradial amputee patients. The Jill prostheses will be available to all patients who need our prosthetic system, regardless of financial background. If the patient cannot pay for the product and services, the Jill prosthetic team will ensure that the prosthesis is funded through financial aid options such as charities and donations.

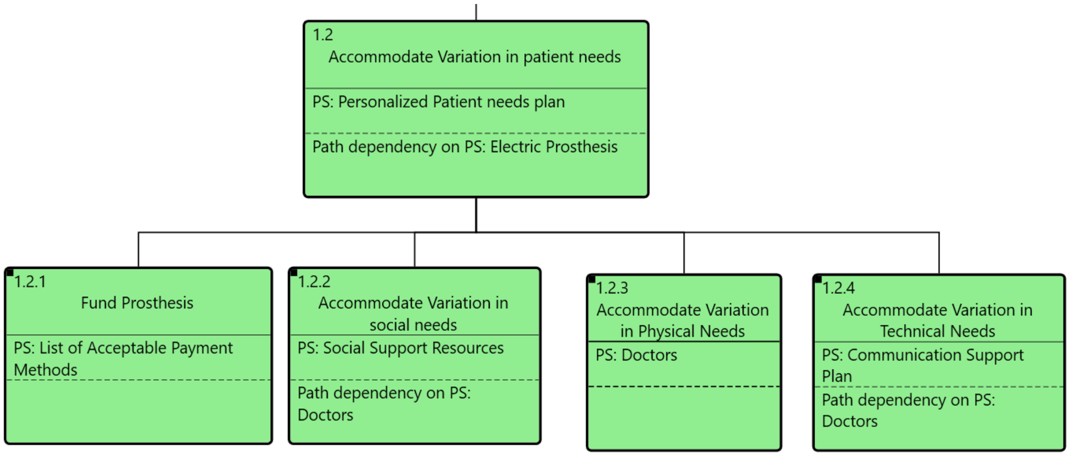

Fig. 6.8. Accommodate variation in customer needs branch

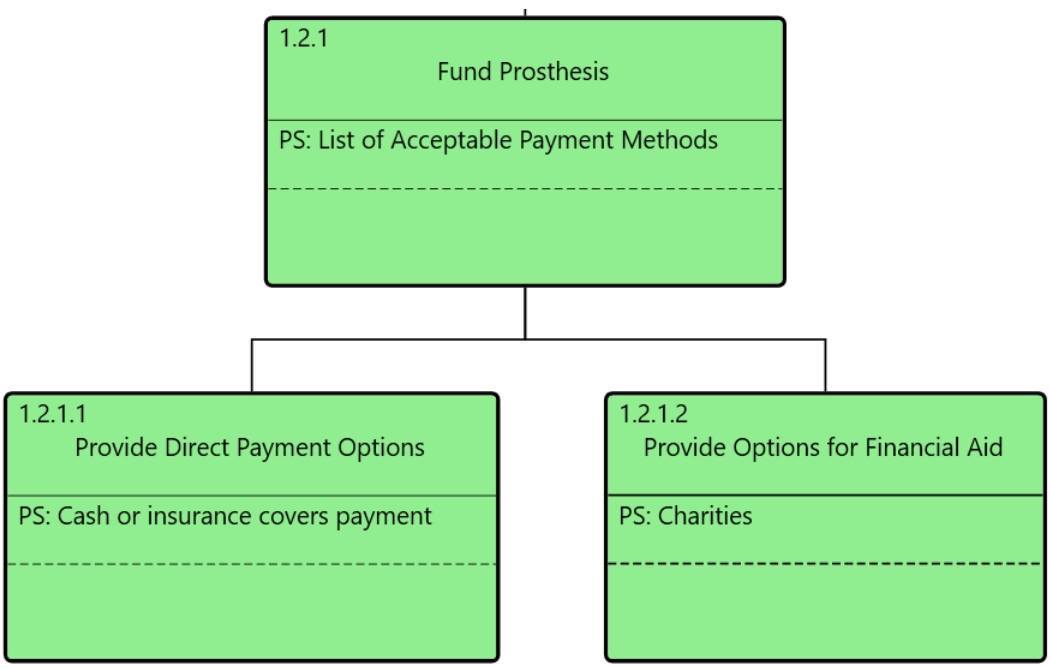

Fig. 6.9. Decomposition of the list of acceptable payment methods

The Jill prosthetic team knows that losing a hand can be a very traumatic experience. To improve the lives of transradial amputee patients, the Jill prosthetic system will accommodate variation in social needs. The decomposition is illustrate in Figure 6.10. The Jill prosthetic system will provide social support resources by facilitating support groups and creating an online presence so anyone, anywhere, can support the patient.

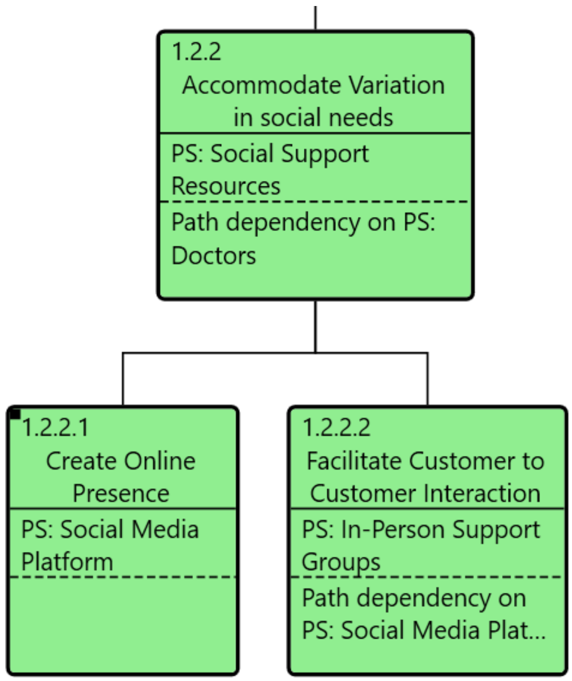

Fig. 6.10. Decomposition of PS Social support resources

The Jill prosthetic system will accommodate the variation in technical needs through a communication support plan. The decomposition of the communication support plan illustrated in Figure 6.11. The goal of this branch is to answer any questions the patient may have and to provide troubleshooting options. If the patient has trouble operating the Jill prosthesis, they have the option to look at our online frequently asked questions page or to call our call center for help. If the prosthesis suffers a critical failure, a technician will travel to the patient to perform repairs. To improve the product, the patient will periodically complete questionnaires related to the usability and difficulties experienced.

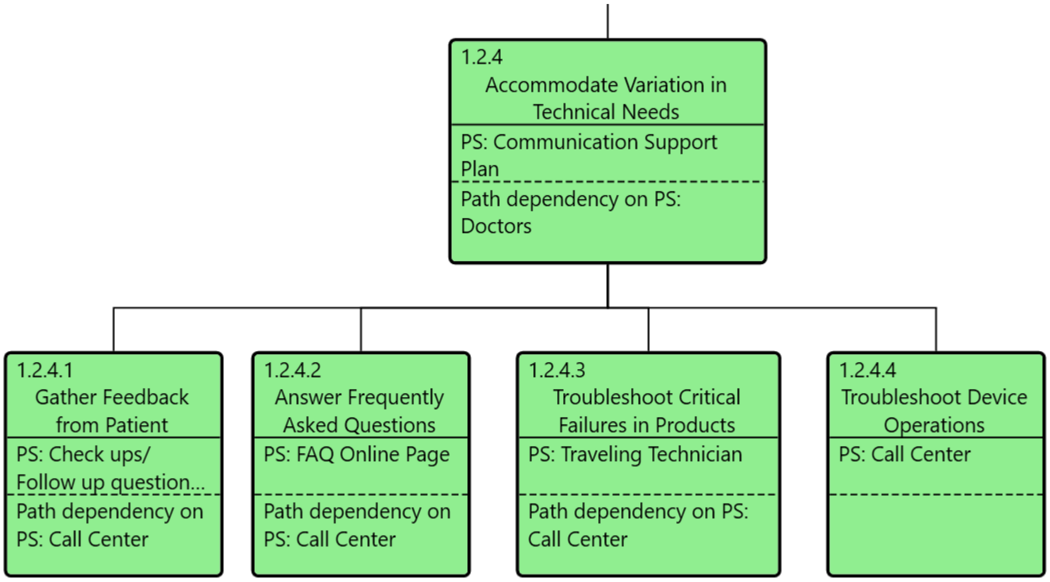

Fig. 6.11. The Jill prosthetic system will accommodate variation in technical needs to ensure the patient is comfortable with the prosthesis

6.7 Testing

The Jill Prosthesis uses a Plan Do Check Act (PDCA) process to continuously improve the design of the product.

Its of utmost importance that each component is traceable to a test activity. Each component and interface within the Jill prosthesis is linked to one or more of the following tests:

- Mechanical actuation test
- Fit test
- Strength of material test
- Grip calibration test
- Electrical component test
- Electrical output test
- Electrical input test
- Delay test
- Battery test
- Thermal balance test
- Life of component test
- Serviceability test

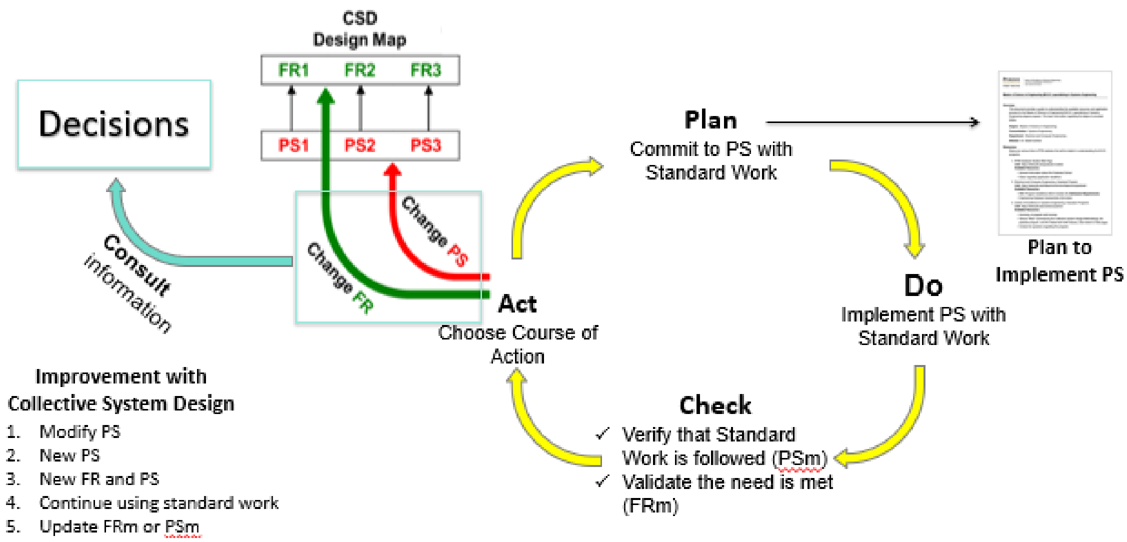

Fig. 6.12. The PDCA is adapted from a continuous improvement process that enables the design to team to make product design changes

For example: the index finger will undergo the mechanical actuation, fit, strength of assembly, grip calibration, durability, component life procedures to ensure all specification are met. The index finger test traceability is illustrated in Figure 6.13.

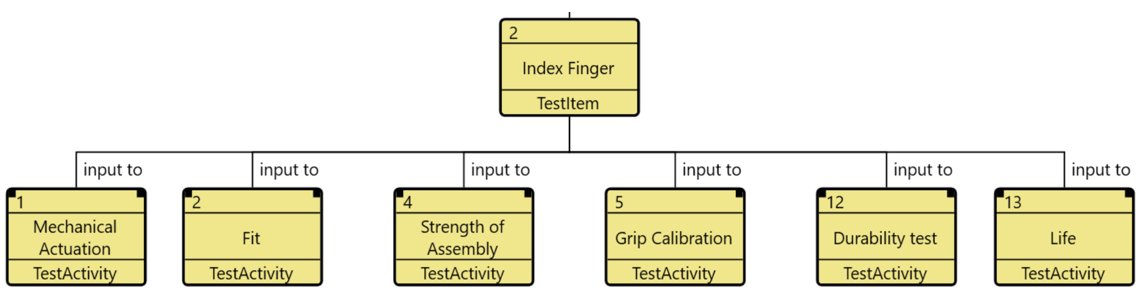

Fig. 6.13. The test activity is traceable to each relevant component and FR

6.8 Design Summary

The Jill prosthetic is designed based on defined customer needs and use cases; this information describes the functional requirements and constraints that make up the Design Decomposition. A predefined decision pattern was used to determine the optimal solution and describe how each FR was defined and how each PS was selected. The current prosthetic design features a custom fitted prosthesis equipped with eight essential grip patterns and five gestures. The Jill prosthesis provides access to support groups and personalized product operation training to enable the patient to regain normal hand functionality. The design is comfortable, affordable, and maintainable while improving overall quality of life for the patient.

The design guide provided a process for the development team to meet all of the system needs. The Collective System Design concepts provided a road map for the design process and separated the system purpose from the means to achieve that purpose. There were many times when team members thought they were describing an FR but were really describing a PS. For example, a team member stated implement checkups and questionnaires as an FR. When asked why we should implement checkups and questionnaires as an FR the team member stated that it was to gather feedback from the patient about the product. By separating what the system must achieve from how the system will achieve it, the design intent was able to be expressed more clearly. The FR was to gather feedback from the patient and the PS was checkups and questionnaires. The development team then realized that the patient would most likely provide valuable feedback to our call center during any troubleshooting phone call. This lead to identifying an additional interface that would not have been possible if the FR had been check ups and questionnaires.

Unfortunately, changes in the design will happen as the design team can not predict the result of every design event. When a design component undergoes a test activity and fails, the development team will need to change a PS or modify an FR or FRm in the design decomposition. The decomposition provided an opportunity for the design team to suggest changes together and see how each change would affect the system. The decision patterns provided a way to explain why an FR was described in particular way and how a PS was selected. The descriptions within the decision patterns enabled the development team to stay aligned with the original design intent and to avoid repeating mistakes.

CHAPTER 7. CONCLUSIONS AND FUTURE RESEARCH

7.1 Research Objectives

This thesis provided a brief overview of traditional design methods and concluded that recognition of common design principles and processes may not be enough to ensure success. The thesis then provides a characterization of design processes and design support to identify social and technical designer needs (Research Objective 1). The thesis then describes a conceptual product design guide which builds on core design concepts from Collective System Design (CSD) and Decision Driven Design (DDD) that would meet designer needs (Research Objective 2). Next, this thesis applies the conceptual product design guide to effectively describe the design of the Jill prosthetic hand system. The Jill prosthetic hand system design focused on improving the lives of transradial amputee patients and to show that the design guide can describe a new and innovative product.

7.2 Research Hypothesis Results

The hypothesis of this thesis (H_A) was that meeting designer needs will improve conceptual product design. The null hypothesis (H_0) was that meeting designer needs will not improve conceptual product design. This thesis provided a direct application to improve conceptual product design and all of the designer needs were met. The Jill prosthetic system met all of its design goal and will improve the lives of transradial patients. The design defects and potential reworks that were avoided due to meeting the designer needs improved the overall quality of the conceptual design of the Jill prosthetic system. This burden of proof provides the argument for why the null hypothesis can be rejected in favor of the alternative.

H_0: Meeting designer needs in Conceptual product development processes will not improve the product design.

H_A: Meeting designer needs in Conceptual product development processes will improve the product design

7.3 Contribution to the Existing Body of Knowledge

This thesis brings together the bodies of knowledge in systems engineering and product development methodologies. This thesis identifies social and technical designer needs within conceptual product development. In addition to identifying designer needs, this thesis provides a guideline to conceptual product design to meet designer needs. This thesis argues that in meeting designer needs that the product design will improve. The guideline represents a potentially valuable viewpoint that will improve product design due to the assertion that design processes should meet both end user and designer needs.

7.4 Lessons Learned

There were two main lessons learned during the design of the Jill Prosthetic system.

- When facing failures and design issues people tend to switch to a solution mode before identifying the root cause.
- The key to successful product design is common communication of design decisions and representations.

The first lesson learned deals with understanding that design is a decision making process. Determining all possible outcomes of design events is near impossible. There will be times when the design has to change. When change is necessary it is best to avoid adjusting the design without consulting the information that lead to original design commitments. Failure to do could lead to future design mistakes as the changes could have unexpected results on other parts of the design.

The second lesson deals with common communication of design. At the start of the design process every team member has their own definition and interpretation of what design should accomplish. Having a common language to communicate design decisions enables to design team to be in agreement of what must be accomplished and how to accomplish the design's goals.

7.5 Future Research

The future research related to this thesis will be focused on applying the proposed guideline to other projects. The additional data gathered will be used to refine the steps and possibly implement additional steps.

The proposed guideline does not currently facilitate a feedback loop from testing to physical components of a design. For example: a design component, modeled in computer aided software (CAD), has failed to fit into the corresponding place in the product. The research should be directed at adding a step between steps 6 and 7 to create a feedback loop between testing procedures and the corresponding Physical Solutions.

REFERENCES

REFERENCES

[1] "Systems Engineering and the World of Modern Systems," in *Systems Engineering Principles and Practice*. John Wiley & Sons, Ltd, 2011, pp. 3–26, section: 1 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118001028.ch1. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118001028.ch1

[2] "About Systems Engineering." [Online]. Available: https://www.incose.org/about-systems-engineering

[3] W. Eder, "Engineering Design: Role of Theory, Models and Methods," Feb. 2014, pp. 193–217.

[4] L. Blessing, "Comparison of design models proposed in prescriptive literature," 1995, library Catalog: www.semanticscholar.org. [Online]. Available: https://www.semanticscholar.org/paper/Comparison-of-design-models-proposed-in-literature-Blessing/a4ebd5a1b0ab1806342524f4ae2892c270e2da0f

[5] P. Badke-Schaub and O. Eris, "A Theoretical Approach to Intuition in Design: Does Design Methodology Need to Account for Unconscious Processes?" Feb. 2014, pp. 353–370.

[6] P. E. Vermaas, "Design Theories, Models and Their Testing: On the Scientific Status of Design Research," in *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*, A. Chakrabarti and L. T. M. Blessing, Eds. London: Springer, 2014, pp. 47–66. [Online]. Available: https://doi.org/10.1007/978-1-4471-6338-1_2

[7] G. Shea, "Systems Engineering Handbook," Feb. 2019, library Catalog: www.nasa.gov. [Online]. Available: http://www.nasa.gov/seh/index.html

[8] 14:00-17:00, "ISO/IEC/IEEE 15288:2015," library Catalog: www.iso.org. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/37/63711.html

[9] "Interface Management," library Catalog: acqnotes.com Section: CareerFields. [Online]. Available: http://acqnotes.com/acqnote/careerfields/interface-management

[10] "Risk Management - SEBoK." [Online]. Available: https://www.sebokwiki.org/wiki/Risk_Management#Risk_Management_Process_Overview

[11] C. Chapman, "Project risk analysis and management—PRAM the generic process," *International Journal of Project Management*, vol. 15, no. 5, pp. 273–281, Oct. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263786396000798

[12] D. W. W. Rovce, "MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS," p. 11.

[13] "JIRA Waterfall Model - Javatpoint," library Catalog: www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/jira-waterfall-model

[14] "CS 460 Software Engineering." [Online]. Available: https://people.wou.edu/~morses/classes/cs46x/presentations/CS460_SE_1.html#/2/1

[15] J. Highsmith and A. Cockburn, "Agile software development: the business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, Sep. 2001, conference Name: Computer.

[16] "Product Methodologies - What They Are and How to Avoid Pitfalls," library Catalog: www.pmi.org. [Online]. Available: https://www.pmi.org/learning/library/product-methodologies-software-development-programs-6529

[17] Hemant M. Patil, Saurabh S. Sirsikar, Nitin N. Gholap, and Pillai HOC College of Engineering and Technology, Rasayani, Raigad, Maharashtra, "Product Design and Development: Phases and Approach," *International Journal of Engineering Research and*, vol. V6, no. 07, p. IJERTV6IS070136, Jul. 2017. [Online]. Available: http://www.ijert.org/browse/volume-6-2017/july-2017-edition?download=17142:product-design-and-development-phases-and-approach

[18] E. Céret, S. Dupuy-Chessa, G. Calvary, A. Front, and D. Rieu, "A taxonomy of design methods process models," *Information and Software Technology*, vol. 55, no. 5, pp. 795–821, May 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584912002285

[19] J. R. A. Maier and G. M. Fadel, "Understanding the Complexity of Design," 2006.

[20] "Figure 1.1: Relationship between design freedom and design knowledge in..." library Catalog: www.researchgate.net. [Online]. Available: https://www.researchgate.net/figure/Relationship-between-design-freedom-and-design-knowledge-in-building-design-projects_fig1_279810934

[21] S. Song, A. Dong, and A. M. Agogino, "TIME VARIATION OF DESIGN "STORY TELLING" IN ENGINEERING DESIGN TEAMS," 2003, library Catalog: www.designsociety.org Pages: 423-424 (exec.summ.), full paper no. DS31_1716FP. [Online]. Available: https://www.designsociety.org/publication/24180/TIME+VARIATION+OF+DESIGN+%E2%80%9CSTORY+TELLING%E2%80%9D+IN+ENGINEERING+DESIGN+TEAMS

[22] R. Valkenburg and K. Dorst, "The reflective practice of design teams," *Design Studies*, vol. 19, no. 3, pp. 249–271, Jul. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142694X98000118

[23] K. Dorst and N. Cross, "Creativity in the design process: co-evolution of problem–solution," *Design Studies*, vol. 22, no. 5, pp. 425–437, Sep. 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142694X01000096

[24] G. Baxter and I. Sommerville, “Socio-technical systems: From design methods to systems engineering,” *Interacting with Computers*, vol. 23, no. 1, pp. 4–17, Jan. 2011, conference Name: Interacting with Computers.

[25] A. Dong, A. Hill, and A. Agogino, “A document analysis method for characterizing team-based design outcomes,” Apr. 2020.

[26] K. W. Ng, *A critical analysis of current engineering design methodologies from a decision.*

[27] P. Mendonza and J. A. Fitch, “Object Based Systems Engineering,” 2011.

[28] D. D. Walden, “Systems engineering handbook : a guide for system life cycle processes and activities,” 2015, library Catalog: www.semanticscholar.org. [Online]. Available: https://www.semanticscholar.org/paper/Systems-engineering-handbook-%3A-a-guide-for-system-Walden/78aa99240173decb30e9cce19243bf278dc5c42a

[29] D. S. Cochran and J. J. Smith, “A Systematic Design Approach to Manufacturing Education,” *Procedia Manufacturing*, vol. 26, pp. 1369–1377, Jan. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2351978918307923

[30] J. J. Smith, S. A. Shah, and D. S. Cochran, “Prevention, Early Detection, and Reversal of Type-2 Diabetes using Collective System Design,” *MATEC Web of Conferences*, vol. 223, p. 01018, 2018, publisher: EDP Sciences. [Online]. Available: https://www.matec-conferences.org/articles/matecconf/abs/2018/82/matecconf_icad2018_01018/matecconf_icad2018_01018.html

[31] D. Cochran, “Enterprise Engineering of Lean Accounting and Value Stream Structure through Collective System Design,” Sep. 2014.

[32] D. Cochran, J. Foley, and Z. Bi, “Use of the Manufacturing System Design Decomposition for Comparative Analysis and Effective Design of Production Systems,” *International Journal of Production Research*, pp. 1–21, Aug. 2016.

[33] D. Cochran and J. Swartz, “Sustaining Improvement through Tone in Collective System Design,” May 2016.

[34] P. Mendonza and J. Fitch, “8.2.1 Integrating System Models around Decisions,” *INCOSE International Symposium*, vol. 23, pp. 213–227, Jun. 2013.

[35] P. Mendonza and J. A. Fitch, “2.2.1 Decision Management (DM) as the engine for scalable cross domain Systems Engineering (SE),” *INCOSE International Symposium*, vol. 22, no. 1, pp. 241–254, 2012, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2334-5837.2012.tb01334.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2012.tb01334.x

[36] F. Mistree, “Review of ‘The Principles of Design’ by Nam P. Suh, Oxford University Press, 1990,” *Research in Engineering Design*, vol. 3, pp. 243–246, Jan. 1992.

[37] “GENESYS System Definition Guide,” p. 44.