AUGMENTED REALITY

3D DESIGN SPACE

NG LAI XING

B.Eng. (Hons.), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR FOR PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE

SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2013

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Ng Lai Xing

03 Jan 2014

Acknowledgements

This research journey is long and arduous with a lot of people who had helped me along the way and I wish to take this opportunity to acknowledge them. First and foremost, I would like to express my utmost and sincerest gratitude to my supervisors: Associate Professor Ong Soh Khim and Professor Andrew Nee Yeh-Ching, for their invaluable guidance, assistance and suggestions throughout this research. I am deeply grateful to Prof. Ong for her constructive critiques and comments, which displayed her enthusiasm and immense knowledge on the research topic, and her patience in guiding and correcting some of my mistakes. I am equally grateful to Prof. Nee for his enlightening feedbacks and suggestions, which brought me to the right track whenever I am lost in my research. Without my supervisors, this research would not have been possible.

I am grateful to NUS Graduate School for Integrative Sciences and Engineering and NUS for providing this opportunity and rewarding me with a research scholarship for my Ph.D. research and study. I would also like to express my gratitude to all the professors who taught me in all the modules and provide me with timely advice and knowledge.

I would like to thank my friends for their support throughout. My friends in the ARAT Lab are always available for assistance, advice and discussion. Special thanks go to Dr. Louis Fong, Dr. Zhang Jie, Dr. Shen Yan, Dr. Fang Hongchao, Dr. Wang Zhenbiao, Dr. Jiang Shuai, Dr. Zhu Jiang, Andrew Yew, Yu Lu, Wang

Xin, Zhao Mengyu, Yang Shanshan and Huang Jiming. My good friends are always there to lend an ear even though they may not understand my research and I would like to express my appreciation for them as well.

Last but not least, I would like to thank my family for all their love and encouragement. I am proud of my parents. Without their love, trust and support, I would not have achieved anything in life. My father has always supported in all my pursuits and is always available to give me advices. My sister and brother-in-law are always there for me and I am really grateful to them. My late mother had always been a pillar of support in my life and I would to dedicate this thesis in loving memory to her.

Table of Contents

Summary

Conceptual design is an important early design stage for the product and development process. It is highly challenging and the designers have to understand the design issues, explore the solution space, generate design solutions, reflect and modify the solutions before evaluating them to arrive at a final concept. 3D models are extensively used in product design but not in the conceptual design stage. 3D Computer-Aided Design (CAD) systems that are used to produce the 3D models can lead to circumscribed thinking, bounded ideation and premature fixation. Augmented Reality (AR) is an emerging technology that merges real and virtual objects in a real environment. AR systems are highly interactive and an AR 3D design space will be able to address the issues with 3D CAD systems and be used for conceptual design.

In this research, the main objective is to develop an AR 3D design space for generating design concepts during conceptual design. The developed system, named Augmented Reality Computer-Aided Design Environment (ARCADE), is an AR design space that allows the users to create the function models and 3D models, and evaluate the functional behavior and ergonomics of the design concept. An intuitive method for generating 3D models using bare hand interactions has been developed. The user can create 3D model using the building block approach, which is similar to playing with virtual LEGOs, and the extrusion approach, which is similar to the creating 3D model with conventional CAD systems.

The function model of the design concept is created by the user in the form of a Product Use Model (PUM). In order to represent the design holistically, a Functional 3D model (F3DM) has been introduced in this research and a Function-Behavior-Structure modeling framework has been developed to create the F3DM from the user-defined PUM and 3D model. The F3DM contains the function model, behavior model, product structure model and the geometrical model of design concept. It can be used to verify the functional and geometrical aspects of the design concept and simulate the function behavior during design evaluation. This is more practical and direct as the user will be testing the design concept with a functional virtual prototype.

ARCADE is able to evaluate and analyze the ergonomics of a design using hand strain detection methodology. The hands of the user are tracked and hand strain incidents can be detected when the user is handling the functional prototype during design evaluation in ARCADE. This will provide feedback to the designer and ergonomics issues with handling of the product can be detected and rectified early in the conceptual design stage.

List of Tables

List of Figures

List of Abbreviations

2D	Two-Dimensional
3D	Three Dimensional
AP	Augmented Prototyping
APM	Archetype Product Model
AR	Augmented Reality
ARCADE	Augmented Reality Computer-Aided Design Environment
ARTM	AR Tracking Module
ARWCS	AR World Coordinate System
BHIM	Bare-Hand Interaction Module
BM	Base Model
CAAD	Construction-at-a-distance
CAD	Computer-Aided Design
CADM	CAD software Module
CAVE	Cave Automatic Virtual Environment
DVESM	Design Verification, Evaluation and Simulation Module
ES	Edge Sensor
F/E	Flexural/Extension
F3DM	Functional 3D Model
FBS	Function-Behavior-Structure
FBSML	Function-Behavior-Structure Modeling Language
FBSMM	Function-Behavior-Structure Modeling Module
FEA	Finite-Element Analysis
FEM	Finite Element Method

FP	Fruit Processor
FPC	Fruit Peeling and Cutting
FS	Fingertip Sensor
GPS	Global Positioning System
GUI	Graphical User Interface
HMD	Head-Mounted Device
HSI	Hand Strain Index
HSV	Hue Saturation Vale
I3DMM	Intuitive 3D Modeling Module
ODE	Open Dynamics Engine
OWL	Web Ontology Language
P/S	Pronation/Supination
PDP	Product Design and Development
PUM	Product Use Model
QFD	Quality Functional Deployment
R/U	Radial/Ulnar
ROI	Region of Interest
TC	Table-top Cleaner
UIC	User Interface Component
VM	Visualization Module
VR	Virtual Reality

1. Introduction

Conceptual design is an important process in the entire product design and development process. It is the starting point of creating a product that addresses the needs of the consumer. It is an exploratory process faced with a lot of uncertainties that have to be addressed before a solution can be generated.

3D models have been used in design since the 1990s, replacing 2D technical drawings as the main medium to embody a design before it is manufactured. It is unambiguous and can be enhanced with high fidelity rendering that makes it look similar to the final product. Analyses can also be performed on it. However, the use of 3D models during conceptual design is largely limited to the communication of the final solution. It is seldom used for concept generation compared to sketching. This research aims to understand the underlying reasons and explore ways for better utilization of 3D models in conceptual design.

Augmented reality (AR) is an emerging technology that combines real and virtual objects in a real environment. 3D models are virtual and implemented in AR systems to interact with real objects, including the human users. In this research, an AR 3D design space is developed that can allow the users to create 3D models during conceptual design and investigate the benefits and limitations of using AR and 3D models for conceptual design.

1.1 Product Design and Conceptual Design

1.1.1 What is Product Design?

Design, according to the “The New Oxford American Dictionary” (The New Oxford American Dictionary, 2005), is a plan or drawing produced to show the look and function or workings of a building, garment, or other object before it is built or made. To design is to devise a plan to create something that has either form or function or both. Design is a highly creative process and the onus is on the designers to come up with something novel. Every man-made object is designed and even natural occurrences can be understood and explained using design. Therefore, it is not an understatement to say that design is and will continue to be an important part of our lives.

Product design is a discipline of design that is mainly concerned with the creation of a product that can be sold for commercial gains. It generally involves needs identification, ideas generation, conceptualization, development, manufacturing and testing of either tangible goods or services. This process usually begins with a market plan and ends with a product that can be sold to others. The product design and development process (PDP) usually consists of various stages of distinct yet sometimes overlapping activities (Ulrich & Eppinger, 2004). Most enterprises have their own PDP to manage their products efficiently. In certain industries where competition is very intense, an efficient and effective PDP can be a competitive advantage for the company in terms of faster time-to-market and more product variety.

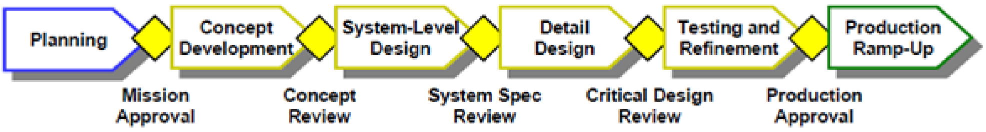


Figure 1.1: Generic product design and development process (Ulrich & Eppinger, 2004)

1.1.2 What is Conceptual Design?

Conceptual design is an early design stage whereby the product concepts are generated and decisions are made on the downstream processes to develop the product. Decisions made during conceptual design have the most impact on the cost of the products produced (Ullman, 2009). Thus, it will be more cost-effective to improve this stage, rather than having efficient downstream PDP processes, such as detailed design, testing and production.

There are different conceptual design definitions. Many design researchers have proposed their definitions of conceptual design and one that is the clearest, concise and relevant to this research is that given by Pahl et al. (2007) as follows:

"Conceptual design is the part of the design process where—by identifying the essential problems through abstraction, establishing function structures, searching for appropriate working principles and combining these into a working structure— the basic solution path is laid down through the elaboration of a solution principle.

Conceptual design specifies the principle solution."

Therefore, conceptual design can be broken down into a series of activities that define and identify the problems and key issues, create and brainstorm the possible solutions, and evaluate and select the best concept for further development (Figure 1.2). It is a highly challenging process that requires both critical and creative thinking and much iteration among the sub-processes. Participants in conceptual design have to think divergently for ideas and ways to satisfy the product requirements derived from market information, and think convergently to combine the ideas to form concepts and solutions. A process of carefully evaluating the product concepts with respect to the requirement lists and design constraints will follow, leading to a selection of the best concept. The conceptual design process is completed after the specification of the final product concept.

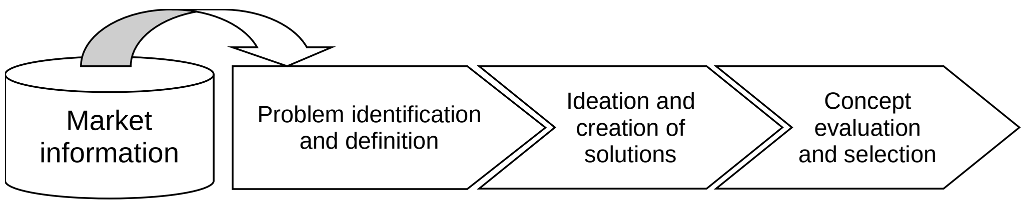


Figure 1.2: Generic conceptual design process

Conceptual design generally revolves around the following four activities:

1. Exploration of the solution space, where the designer thinks of possible solutions that address to some of the required functions of the product and new functions that the product may require.

2. Combination of ideas to form a final solution, where the designer combines different concepts and ideas to form a solution that meets the design requirements.
3. Externalization of ideas, where the designer externalizes the design using a medium, such as 2D sketches, so that the design can be communicated with others.
4. Reflection of the solutions, where the designer reflects and analyses the advantages and limitations of the solutions.

1.2 3D models in Conceptual Design

3D models are generated using 3D Computer-Aided Design (CAD) tools, which allow the user to create and store the design in a 3D data structure. 3D CAD tools are highly efficient in creating geometric representations of product designs and transferring them downstream to the production stage. 3D models are unambiguous and can represent the design in its entirety. The 3D model can be viewed at different viewpoints to develop a complete understanding of its geometry. Technical analyses can be performed on it, such as Finite-Element Analysis (FEA), to simulate how the 3D model will perform under the influence of physical effects, such as force, temperature and aerodynamics.

However, the usage of 3D model is limited to a visualization means during conceptual design. 3D models are not used for exploring the solution space generally and has limited support for the externalization of the ideas and reflection of the solutions, as compared to 2D sketches, which is the dominant medium used for conceptual design. Research on the use of 3D CAD tools has found that they are unable to foster creativity and innovation (Barfield et al., 1993; Robertson & Radcliffe, 2009), and simulate the use scenario to capture tacit user needs. Tacit needs, as opposed to explicit needs which can be obtained by observation and survey and are well documented, are internalized in the users through their memories, experiences and interactions with the product. They are highly experiential and difficult to document.

Some of the problems identified in the limited use of 3D models in conceptual design are:

- A lack of intuitive 3D design generation tools. Conventional 3D CAD tools are more suited for detailed design and value precision, which require the user to define specific dimensions for the 3D models. On the other hand, conceptual design requires design medium, such as 2D sketches, to be generated quickly and can be modified easily.
- A lack of interactive 3D models that can simulate the use scenario to capture tacit needs. The analyses performed on 3D models in CAD tools mainly address the effects of physical phenomena. These are of less concern during conceptual design, where the focus is on generating solutions that can meet the user needs.
- 3D models only represent the geometry of the design. The functions, behavior and structure of the product are defined during conceptual design. 3D models can only be used to present the structure of the product

and are unable to represent the relationships between the functions and geometry of the design.

1.3 Augmented Reality 3D Design Space

Augmented Reality (AR) is a technology that combines virtual and real objects in a real environment. The real and virtual objects will register with each other in real-time and interactively (Azuma, 1997). According to the Virtuality Continuum (Milgram et al., 1994), AR lies closer to the real environment as it uses the virtual to augment the real (Figure 1.3). In an AR system, the boundaries between real and virtual objects are blurred and the users will be able to interact with the real objects that are augmented with virtual objects. This will provide the users with more information of the real objects and enhance the user experience of the real objects. Likewise, interaction with virtual objects is augmented with the use of real objects and the users will be able to experience the virtual objects as though they are real objects.

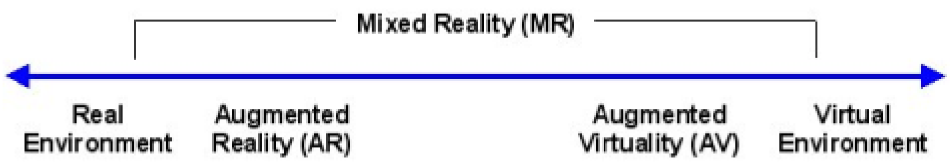


Figure 1.3: Virtuality Continuum (Milgram et al., 1994)

A design space is a set of possible options that meet the objectives and requirements of a specific project given the design parameters that relate to a set of objectives and goals. Exploring a design space means evaluating the various possible design options within a given range of possible products, organization and process designs and optimizing with respect to the objectives and constraints, such as the required functions and costs. This evaluation can be qualitative where the functions and working principles of a product are abstracted in the form of functional block diagrams to evaluate their compatibility, or quantitative where the physical structure and topology of a product are examined to solve the physical constraints, as in a conventional CAD environment.

1.4 Research Motivations

From the preceding sections, it is evident that conventional 3D design systems and tools are unable to support most of the activities for conceptual design for the following reasons:

- 3D design systems bound the users to a workstation and the users have to create the 3D models in a virtual design space, which does not allow them to explore alternative solutions from the one that they are working on currently.
- It is difficult to combine different 3D models to create new solutions due to the precision and completeness of each 3D model. Design features and components of a 3D model have to be modified specifically for each new solution.
- 3D models are an excellent medium to communicate the design solutions of the designer but they are not the preferred tools to externalize a designer's ideas. In order to externalize one's ideas using 3D models, the designer must first know the methods and the steps to create the 3D

models using the 3D design systems. Compared to sketching, this is less intuitive and the cognitive load on the user in creating the 3D models make 3D design systems less suited for externalization.

- Virtual prototypes of the design can be created using 3D design systems for the user to analyze and reflect on the design. However, the analysis addresses only the physical behavior of the product. The functional behavior is more important during conceptual design and this is currently not supported by conventional 3D design systems.

This leads to the following problem statement for this research:

Conventional 3D design systems are not able to support conceptual design adequately, especially during the idea generation and design evaluation processes. There is a need to develop an ideal 3D design system tailored to the requirements of conceptual design.

An ideal 3D design system for conceptual design should allow the user to create 3D models intuitively in the use environment so as to allow the exploration of the solution space and design requirements. In addition, the 3D models must be modular and can be mixed and matched to create alternative solutions easily. Last but not least, the 3D models created should reflect the functionalities and behave like actual products. This will allow the user to understand the design more and be able to select the best solution for further development.

In this research, the main aim is to develop such an ideal 3D design system using a highly compatible technology, namely AR. An AR 3D design system can allow the 3D models to be created in the actual use environment for contextualization. The user can interact with the 3D models in the AR environment and functional behavior of the 3D models can be simulated to reflect the workings of the product.

1.5 Research Objectives and Scope

The developed system, Augmented Reality Computer-Aided Design Environment (ARCADE) is designed to have the following features:

1. Intuitive 3D design modeling, which allows the users to generate 3D models easily using natural interaction tools, such as the hands.
2. Interactive 3D models, which are 3D models augmented with realistic simulation so that they behave like the final products. This allows the users to experience the use of the product before it is manufactured.
3. Design analysis, which provide the users with a better understanding of the product during conceptual design in terms of the ergonomics and the relationships between the designed functions and geometries of the 3D models.

ARCADE is a design environment where real and virtual objects can be manipulated to explore the design issues, create and simulate possible solutions, and evaluate and select the best concept. Users can make use of the actual spatial information in the 3D design space and a mixture of real and virtual objects to design and contextualize new products. Augmented prototypes of the product concepts can be built easily in the design space, with functionalities similar to physical prototypes and flexibility of virtual prototypes. In addition, the product behavior can be simulated and ergonomics issues can be identified during design evaluation in the design space.

The main objective of this research is to develop the ARCADE system and the followings will be achieved as a result of the research:

- Development of an intuitive method for generating 3D models in an AR design environment using bare hand interaction.
- Development of functional 3D models (F3DM) that can reflect the functional behavior of the design in addition to the geometry.
- Development of a Function-Behavior-Structure (FBS) modeling framework that synthesizes the function model, behavior model and 3D model to form the F3DM of the design.
- Development of a design simulation system that allows the F3DM to behave functionally in the same manner as the actual product for design evaluation.
- Development of a design verification mechanism that ensures the consistency between the functional and geometrical aspects of the F3DM.
- Development of a hand strain detection methodology to evaluate the ergonomics of the handling of the 3D models for design evaluation.

This research focuses on the use of 3D models for conceptual design and aims to develop an AR 3D design system that addresses the shortcomings of current 3D design systems for conceptual design. This is achieved by making it easier to create 3D models, allowing the creation of interactive 3D models and facilitating design evaluation of the product in ARCADE.

This research utilizes a design approach in the development of the various features. First, the underlying problems are studied to establish the design requirements. This is followed by a search of possible solutions from existing systems and relevant systems that may be able to address the design issues. Ideas are synthesized to form a solution and this solution will be implemented and evaluated. Refinement and improvements are made to the solution after evaluation and this iterative design process will continue until the solution can solve the problem adequately.

Currently the creation of 3D models on 3D design systems is not as intuitive as sketching. By allowing the user to generate 3D models in an AR environment using his hands, it will be easier to create 3D models and the user can focus on “what to create?” instead of “how to create?”. Sketches and 3D models are not interactive and the functionalities of the product are usually described verbally or literally. The F3DM created in ARCADE is interactive and the user can interact with it directly to understand the functionalities. The functional behavior of the F3DM is simulated and the user can manipulate it like a real product in ARCADE. The design can be evaluated based on its functionality and to a certain extent, its ergonomics.

1.6 Organization of Thesis

The organization of this thesis is as follows.

Chapter 2 will provide literature reviews on the current conceptual design methodologies and tools and the existing and relevant VR and AR design tools.

Chapter 3 will present the ARCADE system architecture and the conceptual design methodology using it. The system setup, hardware and software implemented are described as well. The basic modules, such as the AR tracking module, bare hand interaction module, CAD software module and visualization module are presented in this chapter.

Chapter 4 will describe the intuitive methods for generating 3D models in ARCADE. An earlier work on ARCADE that forms the foundation for the final system will be presented. The general methodology for generating 3D models will be described and two approaches, namely, the Building Blocks approach and the Extrusion approach will be detailed.

Chapter 5 will describe the interactive functional 3D model (F3DM) used in ARCADE and the underlying Function-Behavior-Structure (FBS) modeling framework. A multi-level FBS modeling language has been developed to represent the product and various reasoning methods are deployed to create the F3DM that can represent the product functionally, behaviorally and structurally.

Chapter 6 will cover the design simulation, verification and evaluation that are supported by ARCADE for functional behavior, ergonomics (hand strain) and the functional-geometrical relationships of the F3DM.

Chapter 7 will present three design cases studies that demonstrate the application of ARCADE for conceptual design and Chapter 8 will present the user studies that have been conducted for ARCADE.

Chapter 9 concludes the thesis with a discussion on the research contributions and recommendation for possible future works.

2. Literature Review

2.1 Conceptual Design

In this section, design methodologies that are relevant and highly compatible for use in conceptual design are reviewed. In addition, design tools that support conceptual design will be presented in this section. These tools will be categorized according to the roles they play in conceptual design; their benefits and limitations will be discussed so as to identify the design requirements of ARCADE.

A comprehensive review on design methodologies has been reported by Tomiyama et al. (2009). Among these methodologies, there are three that can be applied for conceptual design, namely systematic conceptual design (Pahl et al., 2007), axiomatic design (Suh, 2005) and total design (Pugh & Clausing, 1996). In addition, a relatively new concept of design thinking (Brown, 2009) can be implemented in this research. In the work by Pahl et al. (2007), conceptual design is broken down into steps consisting of abstracting the essential problems, establishing the function structures, searching for suitable working principles and combining them to form working structures. Axiomatic design (Suh, 2005) uses axioms to analyze the transformation of the customer needs of a product into functional requirements, design parameters and process variables. Total design (Pugh & Clausing, 1996) considers two types of product concepts, namely, static and dynamic, and introduces processes for each concept. In the design thinking model advocated by Brown (2009), there are many interesting concepts and some of the concepts relevant to conceptual design are "converting need to demand", "building to think" and "returning to the surface" (Figure 2.1). These design methodologies will serve as the guidelines on how ARCADE can support the conceptual design process.

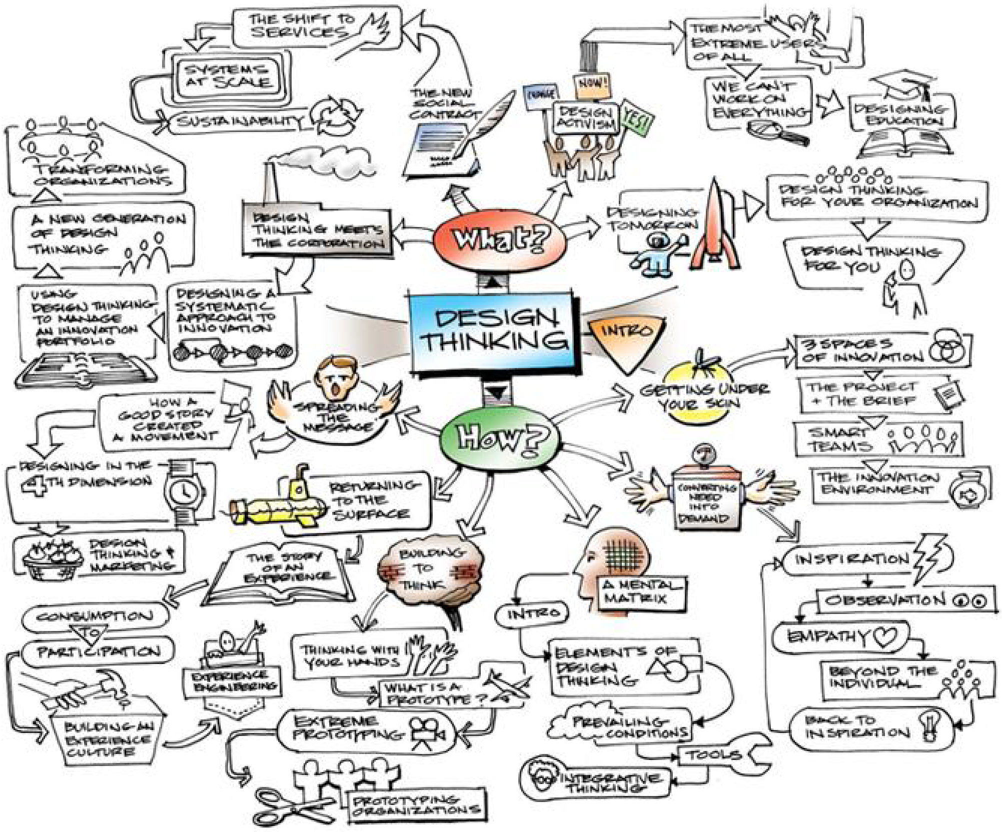


Figure 2.1: Design thinking by Brown (2009)

Design tools that are used in conceptual design can be categorized as follows:

- Market analysis tools
- Idea generation tools
- Concept presentation tools
- Evaluation and selection tools

Market analysis tools are tools that help to identify the problems that need to be addressed. Conducting customers' survey and creating focus groups are two of the most common ways to define the user demands and needs. However, they are only suitable for capturing the explicit needs of the customers, which can only lead to incremental innovation. There is a need to understand the tacit need of the users. The use of observation and empathy is one method to achieve this (Miller & Morris, 1998; Brown, 2009). As quoted from Steve Jobs, "*It's really hard to design products by focus groups. A lot of times, people don't know what they want until you show it to them.*" Therefore, there is a need to create better market analysis tools that are proficient in understanding the explicit and tacit needs. This will be highly beneficial to the later stages of conceptual design as the design requirements list can be formulated according to what the customers really want.

Idea generation tools are tools that help to enhance the creativity of the design team to think of possible solutions and product concepts. Innovation is the key. Pahl et al. (2007) suggested a structured method of decomposing the overall function to many sub-functions; researching and analyzing these sub-functions; and combining them to create new solutions. This is a systematic approach to new ideas generation by dividing and conquering the problems. Brainstorming sessions are commonly used for idea generation. A tried and tested brainstorming method has been presented and practiced with amazing results (Kelley, 2001). It is considered a core competency of IDEO in its position as a world-leading design innovation firm. Another effective idea generation tool is TRIZ (TRIZ, n.d.) which consists of a series of tools and methodologies for generating innovative solutions through the identification and resolution of conflicting constraints. The key idea behind TRIZ consists mainly of identifying the constraints and using analogies of a matrix of known solutions to solve the problems. Creative thinking is very important in conceptual design and more idea generation tools should be used to generate more innovative ideas.

Concept presentation tools are tools that help the design team to share the concepts with others. The concepts can be presented visually using 2D sketches (Lipson & Shpitalni, 2000) and CAD (Robertson & Radcliffe, 2009), and in the form of storyboarding (Sharp et al., 2007) where a use scenario is being described. 2D sketches are usually hand-drawn by designers to provide an image on how a product concept will look when it is realized. Digital 2D sketches done in the computer can also be used, such as Autodesk Sketchbook Pro. Hand drawn sketches are preferred due to the ease of creating new designs and the ubiquity, where one can draw 2D sketches on anything when one thinks of a great idea. This has led to the proverbial term of napkin sketch. The limitations with 2D sketches are that ambiguity is possible due to different perception and views, and not many people are capable of creating good 2D sketches to represent what they think. Artistic talent may be required to create excellent 2D sketches. 3D models created using CAD and 3D modeling software can be used to resolve the ambiguity of 2D sketches. However, they are found to be restrictive for the creative idea generation process (Robertson & Radcliffe, 2009), and a certain level of skill is required to create 3D models. Storyboarding is a way to present the interaction design of a product concept (Sharp et al., 2007). Use scenarios are conceptualized and enacted in front of the target audience to provide them with a better knowledge on how the product concept can be used. The story can be told using placards, posters and video story. AR story boarding is a novel way that is currently being researched (Shin et al., 2005).

Evaluation and selection tools are very established for conceptual design. Some evaluation and selection methods and tools have been presented (Pahl et al., 2007; Pugh & Clausing, 1996). Most of the tools used in the industries are derived from them. Another notable evaluation and selection tool is the Quality Functional Deployment (QFD) which is used to translate the voice of customer to design specifications and then subsequently design decisions. It is a very effective tool which is commonly used in major enterprises, e.g., General Motors and Procter & Gamble. Another form of evaluation and selection tools are decision making tools that actually aim to automate and optimize the decision making process (Vernat et al., 2009).

3D design tools are used for concept presentation mainly to showcase the design in 3D. The 3D models are unambiguous and can be viewed at different viewpoints. This is more efficient than 2D sketches, which requires new sketches for different viewpoints. Analysis can be performed on 3D models to simulate the physical behavior as a form of concept evaluation. However, the results are inaccurate as the 3D models are not detailed enough to include all the features. In addition, the simulation results are less relevant during conceptual design, which has more emphasis on the usage of the product.

Research has been conducted on the use of 3D design tools for idea generation and problem solving (Robertson & Radcliff, 2009; Zeng et al., 2004). It is found that 3D design tools are not utilized widely for idea generation as they result in circumscribed thinking, where the creativity of the design is limited by the software capabilities. The 3D design tools may also bound the idea generation process to the desktop as they can only be used on workstations. Last but not least, 3D models created are detailed and this may lead to premature fixation where the completeness of the 3D model diminishes the need for exploring alternatives. Therefore, 3D design tools must undergo an overhaul before they can be used for idea generation during conceptual design.

2.2 Augmented Reality

2.2.1 VR versus AR

AR is similar to virtual reality (VR) as both technologies create virtual contents that can be perceived by the users. The main difference between VR and AR is that everything in the former is digital, whereas the user can interact with both real and virtual content in a predominantly real environment for the latter. VR is more established than AR and can be considered as a possible solution for this research. Thus, there is a need to compare both AR and VR systems so as to determine which is more suitable as the main technology to be used for the developed 3D design systems.

Proponents of VR systems claim that an immersive VR environment allows the users to be more aware of the information and interact with them in ways which cannot be done physically, e.g., flying through (Segen & Kumar, 1998), dynamic viewing (Kaufmann et al., 2000), and simulates any scenarios that may be hard to replicate in real life (Lin et al., 2008). In a VR environment, anything is possible and things that cannot be done in real life can be replicated in the virtual life. For example, a physically handicapped person will be able to walk and run in a virtual world and perform activities that he cannot do in real life (Wilson et al., 1997). However, the main drawback of VR systems is the inability to support a high-fidelity experience that is close to the real experience. This is due to technical limitations, such as the lack of computational capabilities and image resolution. Besides visual and to a lesser extent audio, other human sensorial systems are not well supported by VR. One cannot interact with a virtual object in the same way as a real one. A virtual flower will only look like its real counterpart and the user cannot smell its fragrance and feel its stalk. In addition, VR systems are very expensive and difficult to set up. Special devices and equipment, such as a head mounted display, data gloves, positional and motion trackers, and Cave Automatic Virtual Environment (CAVE) (Lin et al., 2008) have to be used to interact with the virtual content. As a result, most VR systems are usually standalone systems that are localized in a rigid space, supporting specific well-defined applications. These limit their applications for general use and increase the investment costs for implementing VR systems.

AR is a synergy of the real and virtual worlds, bringing together perception and imagination. It can support the simulation, visualization and modification of virtual objects in VR while preserving the realism provided by the real objects in a real environment. In an ideal AR system, the user will be able to interact with both real and virtual objects in the same manner and view information dynamically while maintaining contextual awareness in a real environment. Interaction tools used for VR can be used for AR with slight modifications. More intuitive tools, such as tangible interfaces and ubiquitous objects (Hong et al., 2008; Duh & Billinghurst, 2008; Irawati et al., 2008) can also be used in AR.

As AR involves the real environment, special setups, such as CAVE, need not be built and potentially any place can be used for AR systems. This makes AR systems highly portable and easily replicable. An example is the LAYAR (LAYAR, n.d.) mobile application which uses the geographical location via Global Positioning System (GPS) and the mobile phone camera to identify the user's current location and field of view to retrieve relevant user-desired information and augment the user's view with this information.

In the context of the research, AR is more suitable than VR due to the following reasons:

- AR can support the use of the real spatial information of the real environment in the design process. Users will have better understanding on the size of the models created by comparing their sizes with those of existing real objects. This is more consistent with the way humans perceive the sizes of objects. In VR systems, the users can only perceive the sizes of the objects created using the numerical dimensions and the existing virtual objects.

- AR allows seamless interaction between real and virtual objects. Real objects can be "modified" easily in AR whereas virtual objects can be contextualized in the real environment to behave as if they are real. In design generation, it will be beneficial for the users to be able to maximize their imagination and create new designs as quickly as possible. Using a mixture of virtual and real objects in AR, unlimited designs can be created. Furthermore, AR allows concurrent design generation, modification, visualization and contextualization in a single environment. It may be possible to replicate some of the contextualization in a VR system; however, prior reconstruction of the real environment has to be carried out, which can consume a lot of time and processing power.
- AR can support direct 3D manipulation of the virtual products and couple modification and visualization of the product. Most established CAD modeling software uses mainly 2D input to carry out 3D modeling operations. This form of interaction is not intuitive and natural, and designers will have to be trained to use such software. The same can be achieved with a VR system but current 3D interaction tools are rather cumbersome. AR offers the opportunity to interact with virtual models with real objects in an unencumbered manner with the use of tangible interfaces and tools.
- An AR system can be portable. The most common and basic form of AR uses computer vision techniques to track and register virtual contents. Simple web cameras can be used for AR applications which greatly increase the portability of the AR systems. This means that any

unprepared environment can be used for *in situ* design in AR. VR requires special devices that are costly and difficult to set up, and this limits the mobility of such VR systems.

2.2.2 VR Design Tools

The use of VR in design has been mainly in the area of computer aided design, manufacturing and engineering. The main purpose of using VR is to integrate the design, manufacturing and testing processes. In Virtual DEsign (Ingrassia & Cappello, 2008), a novel approach of utilizing VR in PDP has been described. VirDe allows the designers to carry out all the design tasks in VR, from modeling to simulation analysis using the finite element method (FEM). It integrates 3D modeling and FEM analysis in a virtual environment supported by a wireless 3D input device. The integration of CAD and FEM analysis allows front-end simulation to be performed when a design is created. Three VR systems, VRAx, NaviMode and ConstructTool are presented for use in design by Weidlich et al. (2009). However, these systems can only make analyzing the design easier and do not enhance innovation in conceptual design. Oh et al. (2006) described a conceptual design system to carry out modeling activities on 3D scenes based on SESAME (Sketch, Extrude, Sculpt, and Manipulate Easily). It makes CAD modeling more suitable for conceptual design. However, due to the use of traditional desktop input like the mouse, the intuitiveness of modeling is limited. As it can be seen from the VR design tools, VR has some inherent drawbacks which make it less suitable for conceptual product design. Therefore, it will be necessary to look at how AR can be applied in PDP and conceptual design, and more importantly in ARCADE.

2.2.3 AR Design Tools

AR can be applied in PDP for design generation, collaborative design, design reuse, prototyping and design visualization. Among these five areas, AR has been applied more extensively in collaborative design, design visualization and more recently prototyping. A comprehensive review of the AR applications in design and manufacturing can be found in a review by Nee et al. (2012).

Some works applying AR in collaborative design are shared-reality meeting (Shared-reality, 2008), tabletop mobile AR (TMAR) (Na et al., 2008) and product information visualization and augmentation in collaborative design (Shen et al., 2008). A general theme of such works is the use of AR to support multiple viewing of a product and annotations and modifications to the reviewed design. This can be done locally in a meeting room or remotely in a distributed setting supported with internet connection.

Some interesting research on the use of AR for design visualization is the Fata Morgana project (Klinker et al., 2002). Webel et al. (2007) reported work on comparing virtual designs with real objects using AR and Weidlich et al. (2008) reported work on product analyses using AR visualization. In the Fata Morgana project (Klinker et al., 2002), virtual car models are overlaid in a real book to provide the users with 3D viewing of a car without having to be physically present in the showroom. This is an AR application that aims to bring the showroom to the customers. In the works by Webel et al. (2007) and Weidlich et al. (2008), virtual models are overlaid on real products to check the differences between the manufactured and the designed product, and to visualize simulation analysis results respectively. AR is used mainly to augment physical products so that they can be evaluated with the computed virtual information.

Augmented prototyping is an emerging field and some interesting research include the augmented reconfigurable foam (Park, 2008), tangible augmented prototyping of handheld digital products (Park et al., 2009), augmented prototyping of information appliances (Aoyama et al., 2009) and work reported by Verlinden et al. (2006). Most of these works use a physical prototype built using rapid prototyping techniques and overlay the virtual product model on this physical prototype. Using these augmented prototypes, the user interfaces and function-behavior of the product can be evaluated on top of the realistic appearances. Simple function-behavior can be simulated even before the hardware and software is ready or available to perform the desired functions. This can be used to test the usability of the product before the design details have been finalized.

AR can be applied in design reuse. Fiorentino et al. (2009) have demonstrated the use of AR on existing technical drawings such that the 3D virtual models can be viewed and manipulated together with the 2D drawings for design reuse. Sidharta, (2006) reported a simple yet effective method for browsing through 3D models using AR markers has been demonstrated. The main use of AR in design reuse is to facilitate information visualization and flow when selecting designs for reuse and testing them with real objects.

Projects that utilize AR for design generation include Spacedesign (Fiorentino et al., 2002), Tinmith (Piekarski & Thomas, 2003), TARM (Park & Lee, 2004), 3DARModeller (Do & Lee, 2008), Napkin Sketch (Xin et al., 2008) and creating freeform surfaces in AR (Fuge et al., 2012). In Spacedesign (Fiorentino et al., 2002), designers can modify the aesthetic design of a car (scaled down) model and create surfaces in a mixed reality design space. This is similar to what ARCADE aims to achieve for design generation; an improvement by ARCADE over this system will be a better integration of both real and virtual objects in creating new models. Tinmith (Piekarski & Thomas, 2003) is a mobile system that performs simple CAD of buildings using pinch gloves and a novel construction-at-a-distance (CAAD) technique and contextualizes them in an urban environment. It is used mainly for mobile urban planning although some aspects of the system can be applied for the design of commercial products. TARM (Park & Lee, 2004) is a system which uses AR and tangible user interfaces to create 3D models. Physical blocks with markers can be manipulated by the user to create his/her desired objects in a manner similar to using building blocks. 3DARModeler (Do & Lee, 2008) is based on 3D Studio Max and can perform the simple operations available in it, such as creating models, adding textures, animations and light sources for the purpose of casting shadows. The system uses the mouse and keyboard, together with AR markers to build 3D models. However, limited modeling operations are supported to modify individual virtual object. Napkin sketch (Xin et al., 2008) uses a UMPC to sketch product concepts and augment them on a planar surface like a napkin, allowing the users to create and share designs. The system reported by Fuge et al. (2012) allows the user to create freeform surfaces in an AR environment using a data glove as the main interaction tool. The five finger tips of the hand are tracked to create a 3D points cloud when they are moving through the air and a pressure sensor is used to control the weight of the points. This allows the user to create freeform surfaces by waving the hand in the air.

All the works that have been discussed demonstrated the benefits of using AR in design generation. However, they are less suitable for conceptual design of consumer products and do not make use of physical objects and constraints in the design process. Therefore, ARCADE will attempt to address these drawbacks by allowing the user to create 3D models using a combination of virtual and real objects with tangible user interfaces in its intuitive 3D design modeling module. Concurrent design, visualization and contextualization can be performed in real time, leading to a more efficient design process.

2.3 Enabling Technologies

This section reviews two enabling technologies used in the development of the AR 3D design system, namely, bare-hand interaction and function modeling.

2.3.1 Bare-Hand Interaction

Bare-hand interaction is less intrusive and more convenient for users to interact with the virtual contents. Earlier AR interfaces that use computer-vision based hand tracking typically track special markers that are attached on the hand and fingers, such as thimble-shaped markers and ultraviolet light sources to detect the positions of four fingertips and support gestural inputs (Kim & Fellner, 2004), colour markers at the fingertips in the SixthSense system (Mistry et al., 2009), and hand-worn gloves with the fiducial markers attached on the thumbs (Piekarski & Thomas, 2003). Using markers is an effective method to simplify the hand feature detection procedure, and the gesture parameters can be calculated efficiently. However, the markers must be specially designed for calibration and tracking as there is a limit to the number of markers that can be placed due to space constraints.

Bare-hand interaction methods can be classified roughly into two groups, namely, gestural and direct manipulation. Gestural bare-hand interaction utilizes vision-based hand detection and tracking systems to identify the gestures of bare hands from video streams and use them as commands, which computers can understand and respond to. Such systems can be used to recognize simple sign language (Nielsen et al., 2004), interact with existing computer applications (Dhawale et al., 2006; Hilliges et al., 2009), navigate object repository mapped in a 3D virtual environment (Chen et al., 2007), and game control (Schlattmann et al., 2009; Yoon et al., 2006).

Direct manipulation bare-hand interaction is triggered when there is a contact between the hand and the virtual objects. 3D hand model-based tracking system track the articulated 3D pose of a hand while the hand is interacting with objects to obtain accurate hand and finger positions (Hamer et al., 2009; Du & Charbon, 2007). Due to the high dimensionality of a user's hand, the 3D model-based hand tracking methods are computationally expensive and difficult to process in real time. Wang developed a bare-hand interaction system that is able to achieve direct manipulation in real time for AR application by tracking only the thumbs and index fingers of both hands (Wang, 2013). 3D pinch operations are used to grab and manipulate objects. This is compatible to the design requirements of the AR 3D design space and can be implemented with a few modifications.

2.3.2 Function Modeling

The use of function models for conceptual design is advocated as a systematic approach to conceptual design (Pahl et al., 2007). Many researchers have developed their own models and reasoning processes, such as Function-Behavior-State modeling (Umeda et al., 1996), Function-Behavior-Structure modeling (Gero & Kannengiesser, 2004), and Structure-Behavior-Function modeling (Goel et al., 2009). Different models and ontologies (Kitamura et al., 2004; Bracewell & Sharpe, 1996) are used to reason the functions of a product. A prerequisite for the use of these models is that they can describe the functions of a product and are decomposable. Reasoning is performed using a divide-and-conquer approach to break down complex functions into simple sub-functions (Goel et al., 2009; Chakrabarti & Bligh, 2001). Many tools have been developed, e.g., commercial tools such as Modelica (Modelica, 2010) and 20-Sim (20-Simi, 2010), and in the academia, KIEF (Yoshioka et al., 2004) and Schemebuilder (Bracewell & Sharpe, 1996), for function modeling and reasoning. Function modeling and reasoning generally describes the transformation of energy, material and signal between the components in a product. As the design of different products has different requirements, a standard function model that can be used in all design scenarios does not exist.

2.4 Requirements of an AR 3D Design Space

From the literature review conducted, the requirements for a 3D design tool for conceptual design, in particular for idea generation and evaluation, can be established as the followings:

1. The 3D design tool must be intuitive and easy to use. This is to prevent circumscribed thinking evident in conventional CAD tools.
2. The 3D models created cannot be too detailed and must be modifiable easily. This will prevent premature fixation and allow alternative designs to be generated easily. Detailed 3D models can be created in conventional CAD tools for concept presentation.
3. The 3D design tool should be portable and can allow the idea generation process to be conducted preferably in the use environment of the product. This will allow contextual inquiry on the tacit needs and more exploration of the solutions and requirements of the product.
4. The 3D models should simulate the usage of the product for design evaluation. This will allow better evaluation of the solutions and concepts

that are less user-friendly and ergonomics can be identified and eliminated. Use issues can also be detected and lead to new requirements as a result.

AR is highly compatible to these requirements. The interactivity and intuitiveness brought about by AR can be used to create 3D models easily and enhance creativity during conceptual design. The contextual information provided using AR can be used to simulate user experience while maintaining the realism of a physical mock-up and the modifiability of a virtual prototype. An AR system is portable and can be set up in the use environment for the idea generation process, where real and virtual 3D models can be used to generate possible solutions.

For the first and second requirements, the design specifications are:

- The system must allow the user to create 3D models faster than with a conventional CAD system
- The number of steps required to create 3D models should be fewer and the steps should be easier to learn.
- The user must be able to modify, mix and match the 3D models on-the-fly so that alternative designs can be generated from existing ones.
- The accuracy of the 3D models needs not have to be very accurate as only the general shape and size are required for conceptual design.

For the third requirement, the design specifications are:

- The system setup should be portable and consist of equipment and devices that are readily available.
- The system should be able to work in both desktop and laptop computers.

For the fourth requirements, the design specifications:

- The 3D models created in the system should demonstrate the functionality of the product on top of the geometry.
- The functions of the 3D models can be defined and modified easily.
- The functions and geometry of the design have to be consistent.
- The system should simulate the functional behavior of the product when it is used. This will provide a more practical evaluation of the concept.
- The system should support the evaluation of the ergonomics aspect of the design by interacting with the 3D models. Evaluation of design ergonomics cannot be supported without the fabrication of the product, which increases the lead time and costs, especially during conceptual designs where there are many solutions that have to be evaluated.

These design specifications define the development of the various modules, in particular the intuitive 3D modeling module (Chapter 4), the function-behavior-structure modeling module (Chapter 5), the design verification, evaluation and simulation module (Chapter 6), of the AR design system created in this research.

The main design constraint for the development of the AR design system is the balance of real-time performance for the various operations and the quantity of information that are processed for the operations. As AR systems require real-time interaction between virtual and real objects, the resolution for this constraint is to process as much information as possible in real-time and if impossible near real-time, which results in a few seconds of delay.

3. The Augmented Reality Computer-Aided Design Environment (ARCADE) System

3.1 Introduction

In this chapter, the conceptual design methodology using ARCADE is presented. ARCADE allows the user to create the function model of the product by defining how the product can be used in the form of a product use model. In addition, 3D models of the design can be created using bare hands as the main interaction tool. The function model and the 3D models will be combined to form a functional 3D model (F3DM) to represent the design and the user can test the product directly in ARCADE and the functional behavior of the product will be simulated. In addition, the ergonomics of the design can be evaluated by detecting possible hand strains when the user is interacting with the product.

The system architecture of ARCADE will be described as well. It consists of seven modules, namely, the AR tracking module (ARTM), the BHI module (BHIM), the intuitive 3D modeling module (I3DMM), the function-behavior-structure modeling module (FBSMM), the design verification, evaluation and simulation module (DVESM), the CAD module (CADM), and the visualization module (VM). ARTM performs tracking and registration. BHIM detects and tracks the hands and fingers of the user and calculates their 3D poses for interactions with the 3D models. I3DMM supports the intuitive generation of 3D models using bare-hand tracking from the BHIM and will be discussed in depth in Chapter 4. FBSMM is used to synthesize the function model and 3D model to form the functional 3D model (F3DM) through rigorous function-behavior-structure reasoning processes. It will be presented in Chapter 5. DVESM is used to verify the design based on the functional and geometrical aspect, provide evaluation of the product based on the ergonomics and user interaction and simulate the behavior of the product when the user interacts with it in ARCADE. It will be described extensively in Chapter 6. CADM provides basic modeling for design generation and constraints information for assembly, as well as detailed design modeling. VM renders the virtual models with the real objects.

3.2 Conceptual Design Methodology using ARCADE

A design concept is a working structure that has functions which can meet the design requirements. Using ARCADE, the users can generate a design concept by first specifying product functions and the user interactions that the product can have via the selection of desired user inputs and product responses using the Product Use Model (PUM); this is followed by creating a basic 3D model of the product (i.e., the form of a design) using their bare hands supported by the I3DMM. The PUM and 3D models will then be processed using the FBS modeling framework in the FBSMM to establish the product's F3DM, which can be evaluated as a functional prototype by the DVESM.

3.2.1 Definition of Product Use Model

A product can respond in various ways based on user interactions with it. The PUM models the user interactions that the designer defines for a product. User interaction is abstracted to consist of the input of the user to a product (user input) and the output that the user will receive from the product as a response to the interaction (product response). The user input is in the form of a user action acting on a user interface component (UIC). The actions can be physical with the user interacting with the UIC physically, or informational whereby the user provides information that is received by the UIC. Product response is in the form of behavioral changes of a component of the product which this component undergoes as a result of the user input. Figure 3.1 shows the general form of product use modeling. For example, a user needs to press a key or move the mouse (physical inputs: Press-Keyboard, Move-Mouse) and enter a password (informational input: Password-Computer) to unlock a personal computer. Based on the input of the user, the computer will either show an unlocked screen (informational response: Screen-Unlocked) so that the user can use the computer or a locked screen and a chime (physical input: Speaker-ErrorChime) due to wrong password.

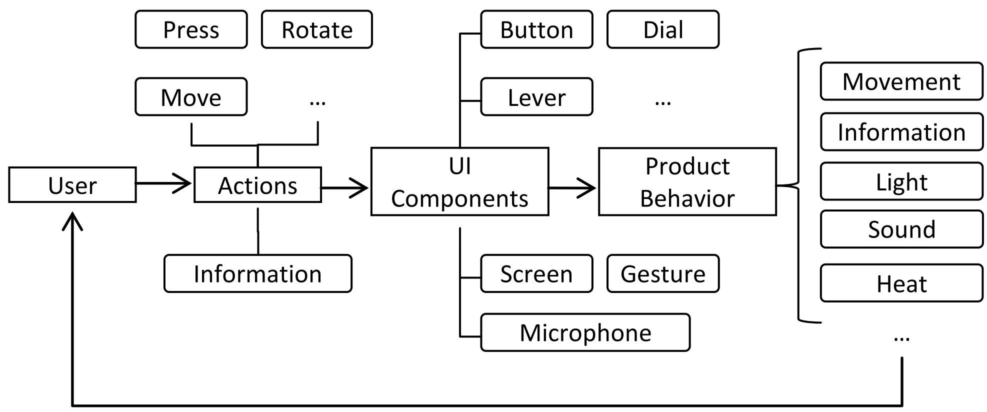


Figure 3.1: Product use modeling

The user defines the PUM of a product by selecting the user action, the UIC involved, the product component that will respond and the change in states of the component. The user can select the UIC from a predefined list of UICs available and the desired user action from a list of mechanical and information inputs that are supported by the selected UIC. When this is completed, the user input will be defined and the user will next define the output in terms of the product response. The component will be selected first from the list of components extracted from a product database followed by the behavioral changes. The lists of UICs, user actions, components and behavioral changes are derived from a database of fifty household devices and appliances.

3.2.2 Generation of 3D Models

The user can create a 3D model of the product by using his bare hands to create and manipulate virtual building blocks in an AR environment. This is analogous to using building blocks like LEGO to create new designs. The building blocks can be modified, oriented and configured. The user can create the desired basic building blocks and combine them together to form the design. With real-time tracking of the user's bare hands and rendering of the models created in an AR environment, the user has a better spatial perception of the design with respect to the real environment. A building block is created by tracking the 3D positions of the fingers of both hands and using the actual spatial dimensions to define the dimensions of the block. There are seven basic building blocks, namely, block, wedge, cylinder, cone, sphere, hemisphere and torus, and they can be used to represent the various components of a product. In addition, design features can be added to the building blocks using conventional CAD functions such as extrusion, sweep and loft. At the end of a 3D design process, the basic assembly configuration of the components is created. Detailed description of the intuitive 3D modeling process will be described in Chapter 4.

3.2.3 Creation of Functional 3D Model

From the PUM and 3D model of a product created by the user, the FBS modeling framework in FBSMM is utilized to reason the functions, structure and behavior of the product and represents them as a F3DM. The functions are obtained from the initial PUM through function reasoning, which decomposes the high level functions from the PUM into FBS primitives and link them to form Function Chains that satisfy the user-defined functions. The reasoned function model of the product is represented as a combination of FBS primitives and Function Chains. The components are arranged in the product structure model based on the relationships they have with other components functionally and geometrically. Functional relationships are derived from the function linkages between the components and they are used to define the type of contacts the components have, which in turn generate a set of geometrical rules which must be fulfilled for the function linkages to be valid. The geometrical information of the product's components is parameterized from the 3D models in terms of their dimensions and the product assembly configuration, and they will be verified with the geometrical rules for design verification. The behaviors are derived from the functions and structure of the product for both expected and unexpected behaviors, which are simulated by matching the required behavior to the corresponding supported simulations to exhibit the behaviors when the user interacts with the product in the AR environment for design evaluation. At the end of the reasoning processes in FBSMM, a functional prototype of the product embodied by its F3DM is created.

3.2.4 Design Verification, Evaluation and Simulation

With the F3DM, design verification can be performed to check whether a design is functionally and geometrically consistent based on the geometrical rules reasoned for the product in FBSMM. Design modifications will be recommended and applied by the DVESM. In addition, the user can test the user interaction and verify whether the product is able to fulfill its intended functions and the possible side effects that the product exhibits. The user interaction will be tracked and possible ergonomics issues involving the handling of the product can be detected so that different design can be evaluated from the ergonomics perspective. The F3DM will be simulated to behave according to the designed functions when the user interacts with it. This allows the user to use the product directly in ARCADE and evaluate a functional prototype that is similar to final product when it is manufactured.

Figure 3.2 provides an overview of the conceptual design methodology using ARCADE.

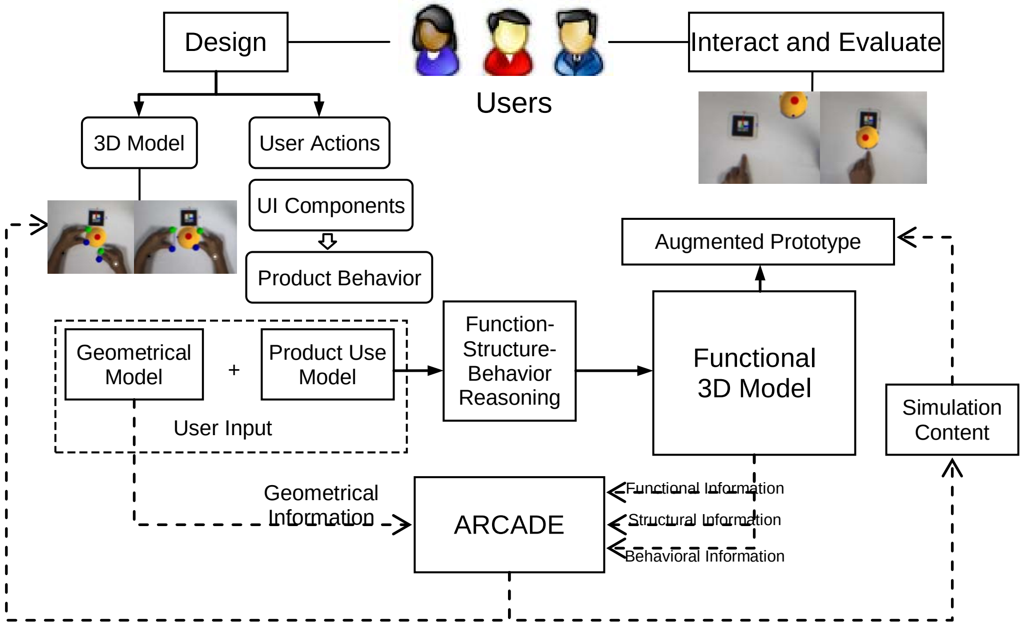


Figure 3.2: Conceptual design process using Functional 3D models in ARCADE

3.3 System Overview

Figure 3.3 shows the system architecture of the ARCADE system. In this chapter, four of the seven modules will be described, namely ARTM, BHIM, CADM and VM.

3.3.1 AR Tracking Module

The main objective of the ARTM is to track various objects in the design environment, such as the hands, existing components and register the virtual models in context so that both real and virtual objects coexist in the AR environment correctly. In order to achieve this, an AR world coordinate system (ARWCS) has been established with the origin at the center of a planar marker that is tracked using ARToolkit (ARToolkit, 2007). The 3D positions of all the objects are referenced from this origin and their relative poses are used to define

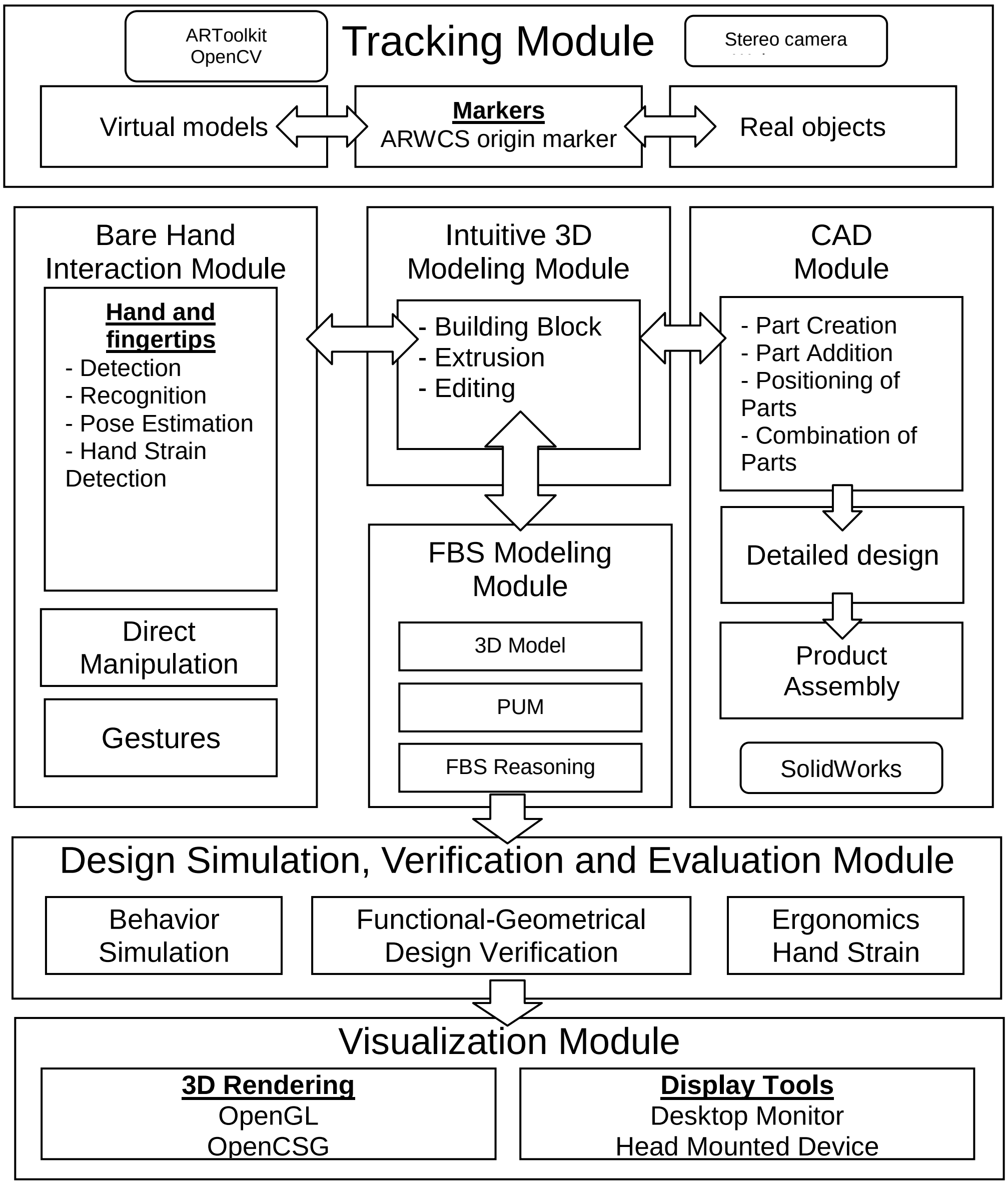


Figure 3.3: ARCADE system architecture

the design and assembly parameters. The origin marker can be placed anywhere in the design environment as long as it can be detected by the camera and the user can move the entire AR world simultaneously for different viewpoints. The origin can be fixed by remembering the last position of the marker and not tracking it to prevent jittering of the 3D models due to tracking failure of the marker. Figure 3.4

shows the framework of the AR tracking module and the relationships of the objects to ARWCS.

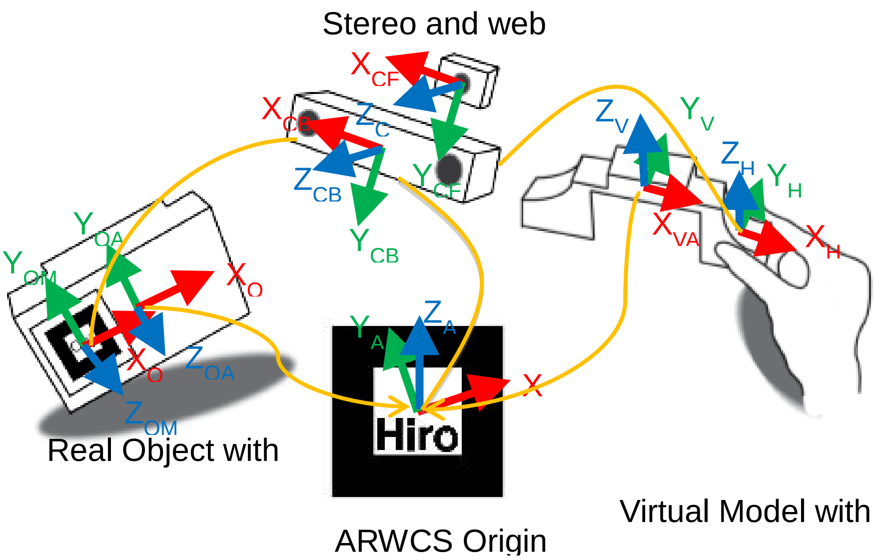


Figure 3.4: Framework of the AR tracking module

Optical tracking is the main form of tracking used in the system and a stereo camera (PGR BumbleBee2) is utilized to capture the 3D information of the scene. The stereo camera is mounted on top of the design space, which dimensions are confined by the field of view of the camera. A design space of approximately 50x50x40cm is used with the camera that is mounted 80cm above the design platform. The camera's view from the top provides a more stable view of the movements of the hands. A second camera is placed near the user's eyes on top of a visor to provide a perspective that is consistent with the user's view. The static stereo camera performs the tracking and the information is relayed to the second camera display.

3.3.1.1 Tracking of Virtual Models

The positions and orientations of the virtual models are referenced from the origin of the ARWCS using their transformation matrices. As the user can manipulate them directly, the poses of the virtual models can be modified based on the tracking of the user's bare hands. The 3D points, representing the thumbs and index fingers, act as control points to achieve bare-hand interactions. Collision between the virtual models and real objects can affect the 3D poses of the virtual models in the design environment.

3.3.1.2 Tracking of Real Objects

Real objects are tracked using markers that are affixed to them. In order to perform collision detection between the virtual models and real objects, the real objects have to be reconstructed as 3D models. If existing 3D models of the real objects are available, they can be loaded onto the markers without rendering them so that the user will perceive that the collision is between the real object and virtual model. In situations where 3D models are not available, users can use commercially available reconstruction software, such as Autodesk 123D Catch. The relative positions of the markers on the real objects with respect to the origin marker can be used to determine the positions and orientations of the real objects in the ARWCS. The relative poses between the object marker and origin marker can be estimated using ARToolkit, and between the object marker and the object are predefined by the location of the affixed object marker. Therefore the relative poses between the object and origin can be derived and the virtual models can interact with the real objects.

3.3.2 Bare Hand Interaction Module

The BHIM detects the hands and the fingertips in the design environment, recognizes the right and left hands as well as the thumb and index fingertips, estimates the poses of the hands and fingertips, and utilizes this information to achieve interaction with the virtual models and carries out different design and product interaction operations. The BHIM is based on the bare hand interaction method developed by Wang in his PhD thesis (Wang, 2013).

3.3.2.1 Detection of the Hands and Fingertips

The hands are detected using the Continuously Adaptive Mean-shift (CamShift) algorithm (OpenCV, 2012). The Hue-Saturation-Value (HSV) color space with the hues separated from the saturation and the intensity is used to create a discrete probability model of the desired hue, representing the skin color of the hand, in the form of a color histogram. A region of interest (ROI) on the hand has to be selected at the start of the process for initialization. Next, the hues derived from the skin pixels in the ROI are sampled and stored into a one-dimensional histogram, which will be used as a reference to detect the skin for subsequent frames. For each frame of the input video stream, the stored skin color histogram is used to convert the image pixels to a corresponding probability of the image using a process called histogram back projection. The CamShift algorithm is used to estimate the hand region based on the probability and shift, resize and re-orientate it accordingly to the hand movements. An assumption made in the implementation of the ARCADE system is that objects with skin color in the captured image are considered to be the hand region, meaning that no other part of the human body can be present in the camera view. After identifying the hand region to be tracked, the hand contour is extracted using OpenCV (OpenCV, 2012), and a distance transformation is performed to find the center of the palm.

With the hands tracked, the next step is to detect the fingertips of the hands as they are the contact points for interaction. The fingertips are detected from the hand contour using a curvature-based algorithm ,Handy AR (Lee & Höllerer, 2007). The curvature of a contour point $C(P_i)$ is measured by computing a dot product of $\overrightarrow{P_iP_{i-1}}$ and $\overrightarrow{P_iP_{i+1}}$ according to Equation (3.1), where P_i is the ith point in the hand contour, and P_{i-1} and P_{i+1} are the preceding and following points respectively, and *l* is the point index on the hand contour which is 15. This means that 15 preceding and following points are used to calculate the curvature of a point. Figure 3.5 shows the vectors used for calculating the curvature. The points with curvature values higher than a threshold are selected as candidates for the fingertips. This will result in the fingertips and the valleys between fingers to be considered as candidates. To differentiate the fingertips and the valleys, the distance between the center of the hand and the candidate points are calculated and the five points with the longest distances are detected to be the fingertips.

$$C(P_i) = \frac{\overrightarrow{P_iP_{i-1}} \cdot \overrightarrow{P_iP_{i+1}}}{\left\|\overrightarrow{P_iP_{i-1}}\right\| \left\|\overrightarrow{P_iP_{i+1}}\right\|} \quad (3.1)$$

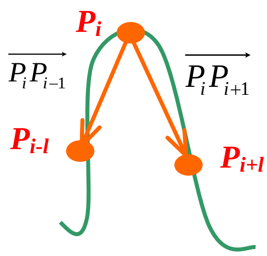


Figure 3.5: Vectors used for fingertip detection

<u>**3.3.2.2 Hands and Fingertips Recognition**</u>

In the BHI module, the tips of the thumbs and the index fingers of the user's hands are used to achieve direct manipulation via a pinching motion. The pinching motion is mainly used for precise manipulation (Feix et al., 2009). Therefore, the system needs to differentiate and recognize the user's left and right hands and thumb and index fingertips automatically.

Hands Recognition

The number of hands in the camera's view can be determined by the number of tracked hand regions. For initialization, the user has to place both hands with the open palm facing down and all ten fingertips visible to the camera. The tip of the thumb for each hand is determined as the furthest fingertip from the mean position of the five fingertips. Next, the direction of the thumb from the center of the hand is calculated. If the thumb is to the right of the center of the hand, the hand is recognized as the left hand and vice versa. This hand recognition method will also work with single hand operation.

Fingertips Recognition

After the user's hands have been tracked and recognized, the thumb and index fingertips on each hand are recognized and differentiated. After the thumb of each hand has been recognized, the index fingertip can be identified as the fingertip that is closest to the thumb. When the user changes to the pinch gesture for direct manipulation, the thumb and index fingertips can be recognized by determining their relative positions with respect to the center of the hand. This can be calculated from the direction of the vector that is the cross product between the vector from the fingertip to the center of the hand, and the vector from the fingertip to the other fingertip. For the right hand, the thumb is to the left of the center of the hand and the index fingertip is to the right. Figure 3.6 shows the results of the hands and fingertips recognition.

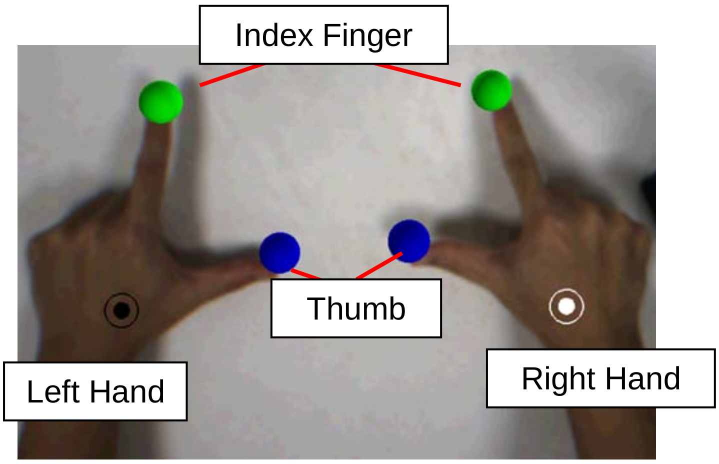


Figure 3.6: Result of hands and fingertips recognition
(Right hand represented by white center point and left hand by black center point; thumb by blue spheres and index fingertips by green spheres)

3.3.2.3 Pose Estimation

With the hands and fingertips detected and recognized, the next step will be to estimate the 3D poses of the fingertips in the ARWCS. With stereo vision, the depth information of the fingertips can be obtained using the disparity information from the two cameras. Therefore the 3D positions of the fingertips are obtained by projecting the 3D information captured using the stereo camera in the camera coordinate system into the ARWCS using a projection matrix.

When the user is manipulating a virtual model directly in the ARWCS, the virtual model will mirror the changes in the translation of the hand. This is done by updating the transformation matrix of the virtual model according to the changes in the translation of the hand, based on the midpoint of the thumb and the index finger T_{hand}. For the interaction to be realistic, the virtual model should rotate according to the hand rotation. To calculate the correct rotation matrix for the virtual model as the hand rotates, a coordinate system is created at the midpoint of the thumb and the index finger using two unit vectors. The first unit vector is between the thumb and the index finger $\hat{V}_{th \to if}$ and the second unit vector is between the midpoint of the first vector and the center of the hand $\hat{V}_{hc \to mp}$. The x-axis of the coordinate system will be the first unit vector, the z-axis is the unit vector of the cross product of the first and second unit vectors and the y-axis is the cross product of the z-axis and x-axis. Figure 3.7 shows the configuration of the coordinate system. When the hand is in first contact with the virtual model, the coordinate system at that point will be recorded as the reference and the displacement of the midpoint to the centroid of the virtual model is recorded as $T_{PMP \to VMC}$. The rotation of the hand $R_{CS_{fc} \to CS_{new}}$ is the rotation from the reference coordinate system CS_{fc} at first contact to the new coordinate system CS_{new} at the new hand position. This can be calculated using Equation (3.2), which first rotates CS_{fc} to the ARWCS, where the x-axis is $\begin{pmatrix}1\\0\\0\end{pmatrix}$, y-axis is $\begin{pmatrix}0\\1\\0\end{pmatrix}$ and z-axis is $\begin{pmatrix}0\\0\\1\end{pmatrix}$, followed by rotation from ARWCS to CS_{new}. The rotation of the virtual model R_{vm} will be a combination of a translation from the midpoint of the pinch to the center of the virtual model, a rotation of $R_{CS_{fc} \to CS_{new}}$, followed by a reverse translation from the center of the virtual model to the midpoint of the pinch and the transformation matrix of the virtual model when manipulated from time, t to $t+1$, $TM_{vm_{t+1}}$, is expressed in Equation (3.3).

$$R_{CS_{fc} \to CS_{new}} = \begin{pmatrix} \hat{X}_{new}x & \hat{Y}_{new}x & \hat{Z}_{new}x \\ \hat{X}_{new}y & \hat{Y}_{new}y & \hat{Z}_{new}y \\ \hat{X}_{new}z & \hat{Y}_{new}z & \hat{Z}_{new}z \end{pmatrix} \times \begin{pmatrix} \hat{X}_{fc}x & \hat{X}_{fc}y & \hat{X}_{fc}z \\ \hat{Y}_{fc}x & \hat{Y}_{fc}y & \hat{Y}_{fc}z \\ \hat{Z}_{fc}x & \hat{Z}_{fc}y & \hat{Z}_{fc}z \end{pmatrix} \tag{3.2}$$

where $\hat{X}_{fc}$, $\hat{Y}_{fc}$, $\hat{Z}_{fc}$, $\hat{X}_{new}$, $\hat{Y}_{new}$ *and* $\hat{Z}_{new}$ *are the unit vectors of the x-y-z axes of* CS_{fc} *and* CS_{new} *respectively.*

$$TM_{vm_{t+1}} = T_{hand} \times T_{VMC \to PMP} \times R_{CS_{fc} \to CS_{new}} \times T_{PMP \to VMC} \times TM_{vm_t} \tag{3.3}$$

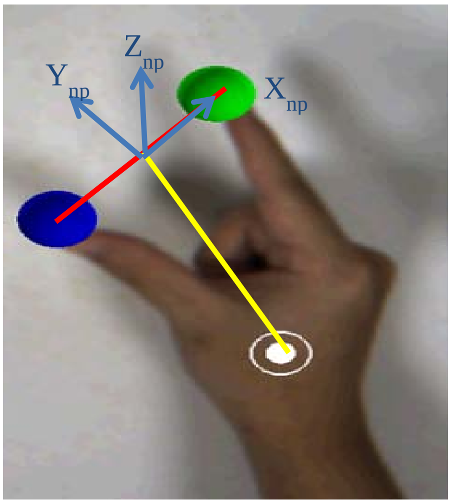


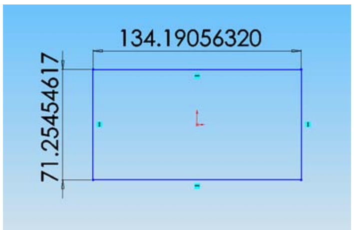
Figure 3.7: Coordinate system used for calculating hand pose

3.3.3 CAD Module

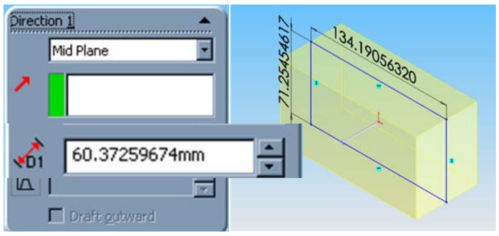
The CADM is used to provide basic modeling support for design generation and design information in ARCADE. It is also used to perform detailed design when required. The CAD model created is informative and can be used to integrate with other design processes, such as physics and dynamics simulations, computer-aided manufacturing, and product lifecycle management. In addition, it is hierarchical, generally represented as a tree from Part-Feature-Faces-Edges-Vertices, and thus it is more comprehensive than a 3D graphics model. Design information, such as the aesthetics and materials, can also be stored in the CAD model.

During design generation, the CAD software performs three supporting tasks, namely, the creation of a part, adding and moving of a part to an existing part and combining parts. A new part is created when the user has created a primitive in the AR environment and defined the dimensions of the part by sizing its bounding box with the hands. Depending on the type of primitive, modeling operations will be carried out automatically using the API. For example, to create a block in the CAD software with dimensions (x_1, y_1, z_1), a 2D sketch of an x_1 by y_1 rectangle is generated and an extrusion of depth z_1 is performed on it. A new part is added to an existing part when the user has added a component to another component in the AR environment. From the relative poses of the parts, the new part is positioned in the existing part accordingly with the relevant translation and rotation. Combination of the parts will then take place based on the Boolean operation that has been set by the user. After combination, the added part will be a feature of the existing part. The CAD model of the combined part will be used to update the surface information of the model in the AR environment. This process will continue until the design has been completed. Figure 3.8 shows the workflow of automatic design creation in the CAD software.

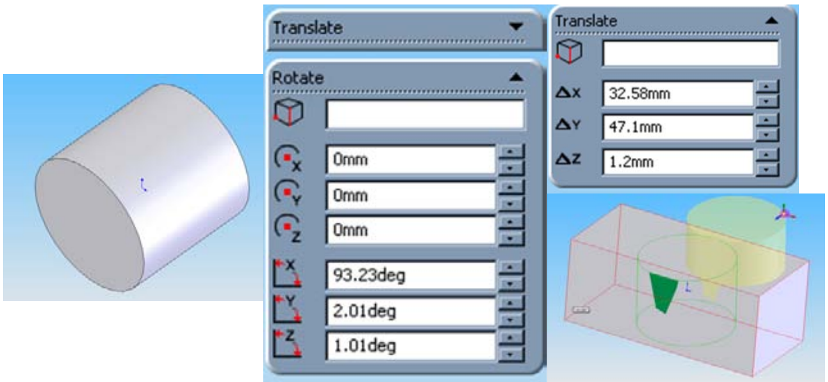
When the design is completed, the CAD software will generate an assembly model based on the parts that have been created. A root part will be identified and the other parts will be assembled onto it based on their geometrical relationships defined earlier. The root part is the component which has all the other components added to it in the design generation process. It is generally the first part that is created. As the completed design is represented by the root part with many other parts added, the root part must be modified before it can be used for assembly. This is done by modifying the Boolean operation from addition to subtraction. This will create the root part with depressions and holes which other parts can fit



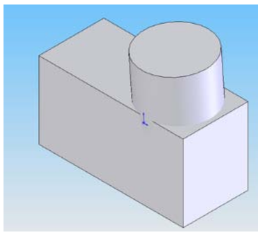
(a) 2D sketch generation



(b) Extrusion of 2D sketch



(c) Addition of a cylinder and placement of part



(d) Combined parts

Figure 3.8: Automatic design creation in the CAD software module

in. An advantage of doing so is that modification of another part during detailed design will also modify the corresponding depression on the root part, which ensures the fit of all the parts. The user can perform modification on the various parts of the design and the design parameters generated by the BHI module in the AR environment. Design features can be added to the parts.

3.3.4 Visualization Module

Visualization is achieved by rendering the virtual models using the OpenGL and OpenCSG libraries and registering them on the markers in the ARWCS. A LCD monitor is used for displaying the virtual objects in the AR environment to allow the user to design in a familiar desktop environment. A Head-Mounted Device (HMD) can be used as the display in the system if the user desires a more coupled modeling and visualization perspective.

3.4 System Setup

3.4.1 Hardware Implementation

Figure 3.9 shows the system setup of the ARCADE system. The system hardware consists of a desktop computer (dual core 2.20 GHz processor, 4 GB SDRAM and 512 MB graphic card), a stereo camera (PGR BumbleBee2), a web camera (PGR Firefly2), a LCD monitor and a HMD (Vuzix Wrap 920).

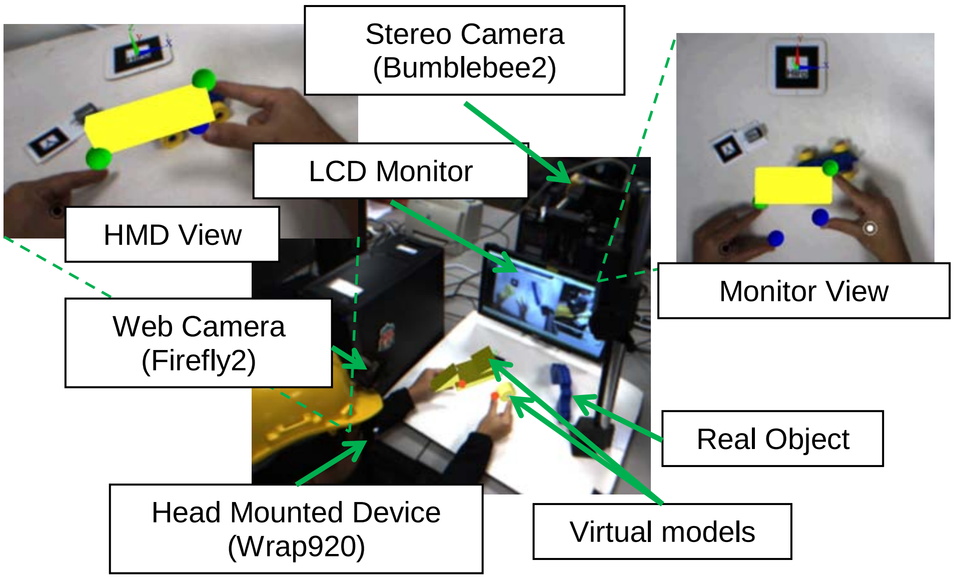


Figure 3.9: ARCADE system setup

3.4.2 Software Implementation

The system is developed in C/C++ using Visual Studio 2008, and open source APIs and libraries, such as OpenCV (OpenCV, 2011) for image processing, ARToolkit (ARToolkit, 2007) for marker tracking, FlyCapture SDK for stereo imaging (Point Grey Research), SolidWorks API (SolidWorks API, 2012) for CAD modeling, OpenGL (OpenGL, 1997) and OpenCSG (OpenCSG, 2010) for 3D rendering and behavior simulation, and V-Collide (V-Collide, 1998) for collision detection. The function-behavior-structure modeling framework is implemented using the Protégé OWL API (Protégé, 2012) in Java and the Java Native Interface API is used to communicate the data between Java and C++.

4. Intuitive Generation of 3D Models in ARCADE using Bare Hand Interaction

4.1 Introduction

The role of design generation tools is to externalize the ideas that the designers have in their mind so that they can share and communicate with others. In addition, by externalizing their ideas in a tangible form, the designers can reflect on them and explore more solutions.

A tool can be described as intuitive if the user can use it based on what one feel is correct without conscious reasoning (Dictionary definition of “intuitive”). A prime example of an intuitive design generation tool is 2D sketch. Designers are generally aesthetically inclined and can draw very well. Therefore, it is natural for them to externalize their ideas in the form of 2D sketches by drawing out how they think the product should work. Research has been conducted on the effects of using 2D sketches for conceptual design and they have been found to be intuitive to the designer, reflective where the designer can look at their sketches and think of improvements, explorative where the designer can start with a random sketch and arrive at a final solution and communicative where the design can be shared with others just by showing them the sketches. As a result, 2D sketches are the dominant design generation tools used for conceptual design.

Conventional 3D CAD tools have a steeper learning curve compared to 2D sketches. In addition, the design generation method is rigid compared to the fluidity of sketching. Certain steps have to be followed before the final 3D model can be created. In order for 3D models to be comparable to 2D sketches as an intuitive design generation, the intuitive 3D modeling module (I3DMM) developed in ARCADE has the following features:

- Bare hand as the main interaction tool for design. The hands are the most intuitive tool that humans have and are used to perform most of the tasks in daily life.
- Direct manipulation of the 3D models created using the hands. The hands are used to handle many real objects by manipulating them directly. The same interaction technique is replicated for the virtual 3D models in ARCADE so that the user interacts with both real and virtual objects in the same manner.
- Familiar design generation techniques that are used to construct artifacts are implemented. The 3D models are generated in a manner that is similar to using building blocks to construct buildings. This is intuitive as most people have some experience playing with building blocks toys and children have no problem with knowing how they are played.
- Editing and design enhancement support are provided to increase the fluidity of the generated 3D models. Conceptual design is an iterative process and there are many modifications to be performed on the 3D models before a final solution can be derived.
- Integration of 3D models created in ARCADE with conventional CAD tools. While it is easier to create 3D models in ARCADE for conceptual design, the later stages of design still require a conventional CAD tool to

generate 3D models that can be manufactured. By integrating the 3D models, detailed design can be performed directly on the 3D models and increase the efficiency.

4.2 Earlier Works on ARCADE

ARCADE begins as an AR design system which uses tangible AR markers to create 3D models. New designs are generated by creating new components, modifying existing components and/or combining these new and existing components. There are two methods of generating 3D virtual models in ARCADE, namely, (i) virtual creation using tangible markers, and (ii) reconstruction and feature extraction from real objects.

A typical modeling scenario for users of the ARCADE system is as follows. Firstly, the users create a virtual base model (BM) either using primitive objects selected from the GUI menu screen, or reconstruction of a desired real object. Next, features are added to the BM. Features can either be created virtually or extracted from the physical features of a real object. This is followed by an iterative process of manipulating the BM and editing the features until the model has been completed.

4.2.1. Creation of Virtual Models

Virtual creation of 3D models involves the use of tangible AR markers. A virtual BM can be created from a group of five pre-defined basic objects, namely, a block, a wedge, a cylinder, a sphere and a hemisphere. Once a desired object is chosen, the user can create the BM using two flat markers. By changing the position of one marker relative to the other marker in the 3D design space, the user can change the size of the BM intuitively. This approach allows the user to make use of the physical space to gauge the dimensions, giving him/her a better perspective of the spatial characteristics. After the BM is created, it will be re-oriented on the flat marker and this marker can be used to position the model in the 3D design space.

Virtual features can be created and added to the BM by first selecting a basic shape. The basic shapes available are rectangle, square, triangle and circle. The size of the basic shape can be determined by either the absolute displacement value or scaled using the relative displacement of the markers. After obtaining the desired 2D shapes of the features to be added, the 2D shapes will be attached to one of the markers so as to select the face and position on the BM whereby the features will be added. To facilitate this process, visual feedback of a change in the color of the sketch will occur when the marker is 'touching' the base model. When the position of a feature is fixed, extrusion or cut-extrusion operations can be performed using the profile of the 2D shape. The extrusion or cut-extrusion will be carried out in the direction normal to the face and the extrusion depth will be the displacement of the marker from the selected face of the BM.

4.2.2. Modeling of Real Objects

Real objects can be reconstructed, created and/or modeled as virtual 3D models based on the captured images of these objects from the web camera. The object of interest is placed in the 3D design space and ARCADE will reconstruct this object using the 2D images captured and the information input by the user about the point-of-views (i.e., top, bottom, front, back, left and right) of the captured images. At least two images with known point-of-views are required for the reconstruction and the user will segment the object of interest from the captured images by adjusting the image threshold. Reconstruction is achieved using a voxel-coloring method based on the information of the outlines of the object at known point-of-views. Figure 4.1 shows the reconstruction of a speaker. The time required for this process depends on the level of details required of the reconstructed object. After reconstruction, a virtual 3D model will be overlaid on a marker and displayed for verification. Once verified, the model will be saved. Features can be extracted using the same method by removing the unwanted parts of the reconstructed models during image segmentation. This method of reconstruction requires minimal input from the users as only two images need to be captured and the respective point-of-views indicated. In addition, 2D sketches can be used to reconstruct the 3D models using this method, eliminating the need of real objects. This enables the users to create 3D models simply by drawing 2D sketches, which is faster and more intuitive.

4.2.3. Modification and Combination.

New designs can be created based on the virtual 3D models that have been created using the AR markers and/or reconstructed and extracted from the real objects. 3D models that are created virtually and reconstructed can be manipulated using the AR markers. The user can modify the dimensions, orientations and positions of the 3D models, and combine them by placing them together in the 3D design space, and selecting the desired Boolean operators and the “Paste/Combine” command. Table 4.1 summarizes the modeling operations.

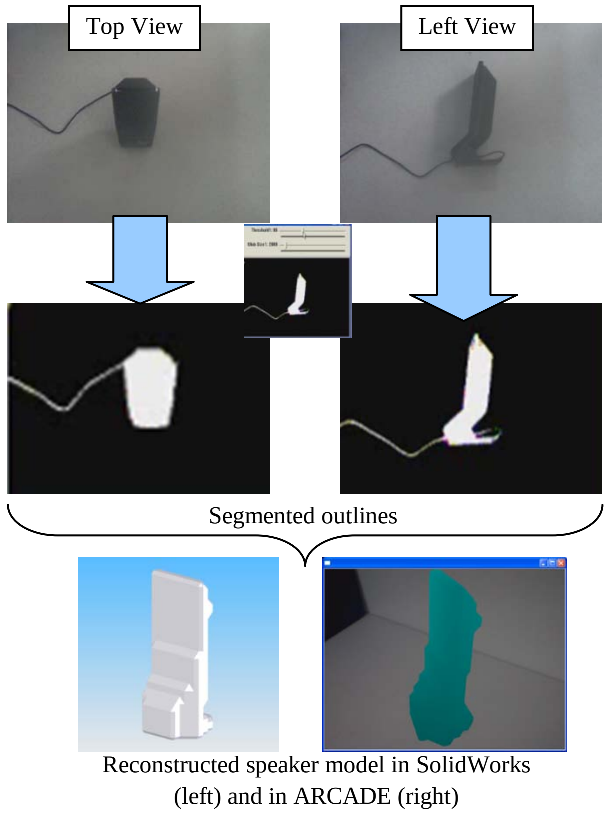
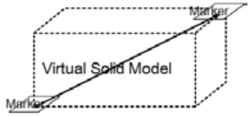
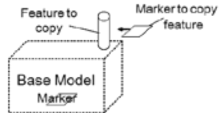


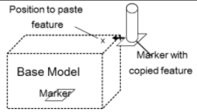
Figure 4.1: Reconstruction of a sound speaker

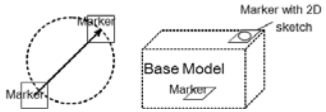
Preliminary user studies have been conducted and will be detailed in Chapter 8. The solid modeling method is found to be simple and fast compared to conventional 3D CAD software. The use of tangible markers provides 3D input information and is more compatible to the modeling operations for creating 3D models as compared to 2D input tool, such as the mouse. The ability to perform the design in a real environment allows contextualization.

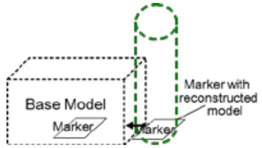
Table 4.1: Modeling operations and corresponding interactions in ARCADE

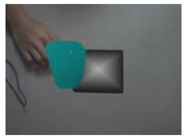
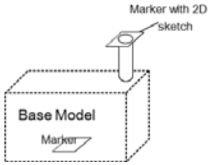
<table>
<tr><th colspan="2">Modeling Operations</th><th>Interactions</th><th>Screenshots</th><th colspan="2">Modeling Operations</th><th>Interactions</th><th>Screenshots</th></tr>
<tr><td rowspan="2">Create base model</td><td>From virtual objects</td><td>

Move 2 markers in the design space.</td><td></td><td rowspan="4">Copy and paste</td><td>Copy feature</td><td>

Move marker near desired feature and copy.</td><td></td></tr>
<tr><td>From real objects</td><td>Place object in the design space. Capture at least two images of it at different point of views for reconstruction. Segment the objects from the background and reconstruct the model</td><td>See Figure 4.1</td><td>Paste feature (virtual)</td><td>

Move marker with copied feature to the desired position to paste the feature.</td><td></td></tr>
<tr><td rowspan="2">Add features using virtual models</td><td>Create and add 2D sketch</td><td>

Modify the size of the 2D sketch by moving 2 markers. Add 2D sketch to the base model by positioning the marker containing the 2D sketch.</td><td></td><td rowspan="2">Paste feature (real objects)</td><td rowspan="2">

Move marker with reconstructed model to desired position and paste.</td><td rowspan="2"></td></tr>
<tr><td>Extrude /cut-extrude</td><td>

Move the marker with the 2D sketch in the direction of the extrusion.</td><td></td></tr>
</table>

The tangible markers have their limitations as factors, such as lighting conditions and shadows, can affect tracking results and jittering is quite common. These limitations affect the accuracy of the 3D models created in ARCADE. The modeling operations are limited and more sophisticated operations should be added. The interaction method can also be more intuitive. All these lead to the second generation of the ARCADE system, which uses bare hand interaction as the main interaction tool and has more modeling operations, using techniques that are familiar to the users, such as building blocks and extrusion processes.

4.3 Bare-Hand Interaction in Design

In Chapter 3, the method for tracking the hands of the user has been described. The tracked information of the hands is used to perform interactions with the system to achieve modeling operations. This section will describe these interactions. Two types of interactions are supported by the BHIM, namely, direct manipulation and gestures.

Direct manipulation allows the users to interact with the virtual objects in the same manner as they interact with everyday objects. Virtual spheres are augmented on the thumb and index finger of each hand. Collision detection between the spheres and the virtual model are performed to check if they are in contact. The color of the virtual spheres will change when they are in contact to provide a form of visual feedback. A collision detection library has been implemented to detect collision and return contact information, such as the locations and the number of contacts.

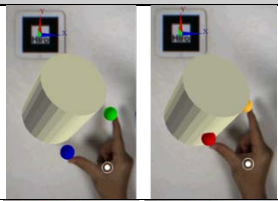
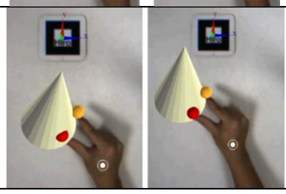
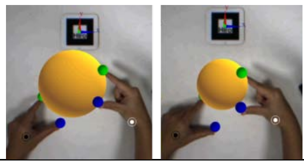
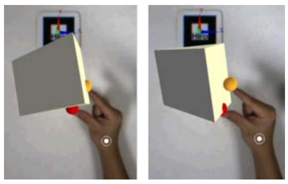
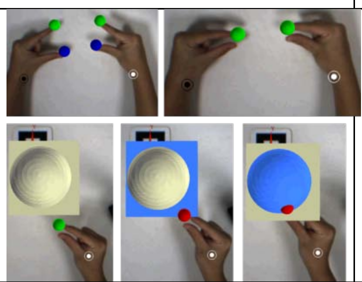
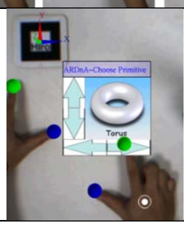
When two spheres (thumb and index finger) from the same hand are in contact with a virtual model, this indicates that the hand has grabbed the virtual model, and the position and orientation of the virtual model will be updated according to the changes in the hand pose. The poses of the fingers and hands are also used to define the dimensions of the virtual models. In general, translations of the hands are used to define the positions and dimensions of the virtual models.

Gesture inputs are used to trigger commands and indicate actions. Two types of gestures are supported, namely, the pinch gesture and the point gesture. The distance between the thumb and the index finger is used to define the pinch gesture. When this distance is below a certain threshold, a pinch gesture will be recognized. The pinch gesture is used as a command input to confirm an action during 3D modeling, and to select a feature of the model so that confusion between grabbing and selecting can be eliminated. The point gesture is achieved with the index finger, and it is used to control a cursor to interact with the virtual panel GUI during modeling operations. Table 4.2 shows the different bare-hand interactions supported.

4.4 3D Modeling with Bare Hand Interaction

Based on the interactions that can be performed by the user's bare hands, 3D models can be created in ARCADE using two types of modeling approaches. The first modeling approach is the building block approach where the user creates building blocks and combines them to form the 3D model of a design. The second is the extrusion approach, which involves creating a 2D sketch first, followed by an extrusion to form the 3D model. In general, the system will detect and identify the hand actions of the user and match them to the corresponding modeling operations in Table 4.2.

Table 4.2 Bare hand interactions supported in ARCADE

Hand Interaction		Tracked Features		ARCADE Operations
Direct Manipulation	Grab	Finger and virtual models		To gain control of virtual models for transformation operations.
	Move	Hand movements		To move virtual models in the design space.
		Finger movements		Movement of fingers to determine the size of BB primitives.
	Rotate	Two quaternions of the fingers with respect to the center of the hand		To rotate the virtual models.
Gestures	Pinch	Measured using a threshold for the distance between index finger and thumb		(i) Command input to confirm actions. (ii) Point selection of virtual models and features.
	Point and Click	3D position of index finger		To act as a cursor and select options on GUI.

The 3D models rendered will change according to the hand actions in OpenGL during a modeling operation. When the user has completed a modeling operation, the design parameters captured by the system will be sent to the CADM to create the 3D model in the CAD software as described earlier in Section 3.3.3.

4.4.1 Building Blocks Approach

The Building Blocks approach for modeling is analogous to using building blocks like LEGO to create new designs. A set of primitive objects can be created, manipulated and combined by the users to generate new designs.

A building block is created by tracking the 3D positions of the fingers in the pinch gesture of both hands and using the actual spatial dimensions to define the dimensions of the block. There are seven basic building blocks, namely, block, wedge, cylinder, cone, sphere, hemisphere and torus. A primitive is created by tracking the 3D poses of the fingers of both hands to define its dimensions and using the pinch gesture to confirm the creation. The first building block will be the base block, and other building blocks are added to this base block to create the final design. New blocks are created using the same method and each block can be manipulated with both hands to define their 3D positions and orientations. Direct manipulation of the virtual models is more intuitive as compared to using a mouse. The user can combine the blocks to form the basic shape of a design by placing them in the desired configuration. When the user is satisfied with the placement of the blocks, the blocks are combined using Boolean operations defined by the user. The positions of the building blocks and the type of Boolean operation will be sent to SolidWorks for the required CAD operation to be performed to create the 3D model. When the CAD operations have been completed, the base block will be updated and the added block will become its feature. More feature blocks can be added to the base block until the design is completed. Figure 4.2 shows the modeling process using the building block approach.

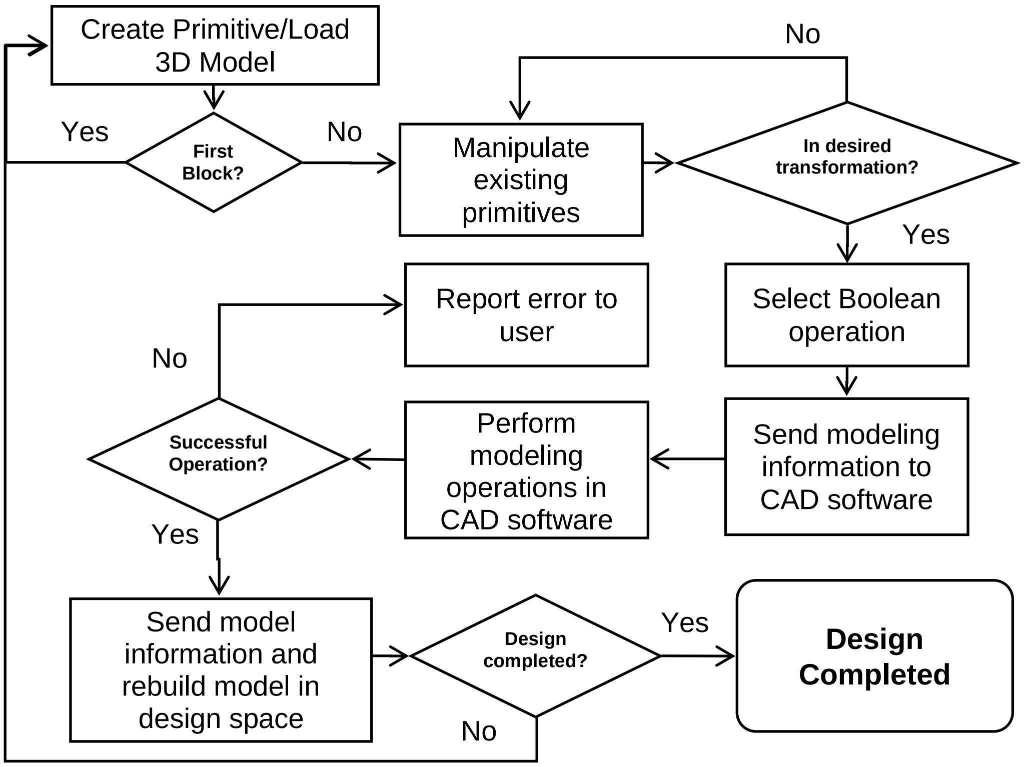


Figure 4.2: Building block modeling process

4.4.2 Extrusion Approach

The extrusion approach is similar to conventional CAD extrusion from 2D sketches in CAD software like SolidWorks and SketchUp. A variety of extrusion operations, in terms of the extrusion path and changing profiles, is supported. In a typical extrusion operation, the user will define the 2D profiles to be pulled and the extrusion path. Using extrusion functions, such as “Extrude” and “Loft” in the SolidWorks API, the user can define both the profiles and paths dynamically to create the intended design.

In general, the user begins an extrusion operation by selecting a face of the model for the feature to be extruded. This is done by touching a face in the model in the pinch gesture to select it. The selected face will be highlighted. Faces that are not in the view of the camera can be selected as the 3D positions of the fingers are captured. After a face has been selected, the position of the 2D profile on the face can be defined by moving the hand touching the selected face to the desired position. This position is determined as the midpoint between the thumb and index finger of the hand. After the position of the sketch has been defined, the path of the extrusion can be defined by moving the hand away from or into the face to specify the depth and direction of the extrusion, and changing the distance between the index finger and the thumb to specify the size of the 2D profile.

Many types of extrusion can be performed in ARCADE by selecting the type of profile to extrude, changes in the profile during extrusion and the type of extrusion path. The user must define these options before the extrusion operation is performed in ARCADE so that the system will be able to detect the hand interactions for various stages of the extrusion operation.

Three types of 2D shapes are supported namely, rectangle, triangle and circle. Freeform 2D profiles can be added by sketching the profile on a flat surface, after which they can be recognized and stored in the database. When the shape of the profile has been selected, a 2D sketch of the profile will be displayed on the position and the surface where the extrusion will take place.

The profiles can be changed in size and type during the extrusion operation. By default, the profile of the extrusion is constant. The user can choose to change either the size or type of the profile to create a more sophisticated feature. For extrusion operation with changes in profile, the user must choose among using a single profile defined at the start point, two profiles defined at the start and end points, and multiple profiles defined along the extrusion path. The distance between the thumb and the index finger determines the size of a profile when the extrusion is performed and changes in size will be recorded accordingly. For changes in profile type, the user must indicate to the system where the profile type should be changed using a pinch gesture and the system will prompt the user to choose the new profile type.

There are three types of paths that can be extruded, namely a normal path, a single directional extrude-to-point path and a freeform path. A normal path extrudes the feature from the profile in the direction parallel to the normal of the surface that the profile is on. The depth of the extrusion is determined from the normal distance between the finger and the surface. A single directional extrude-to-point path extrudes the feature directly to a 3D point in the design space, which is defined by the 3D position of the finger. A freeform path is extrudes the feature in a freeform path that is defined by tracking the movement of the user’s hand throughout the extrusion process. The start point, end point and points in every five frame are used to create a spline that represents the freeform path.

In the most basic constrained extrusion, the user can only pull a fixed 2D profile in a fixed direction that is normal to the 2D profile. Conversely, the most extensive freeform extrusion operation that can be performed by ARCADE involves multiple profiles along a multi-direction extrusion path. Figures 4.3 and 4.4 show the different types of extrusion operations.

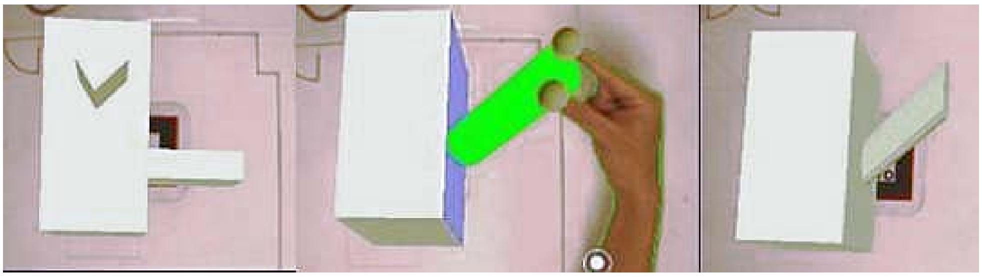


Figure 4.3: Extrusion operation with different profiles and directions
(From left to right: Extrusion from rectangle and triangle profile, defining extrusion direction, extruded feature)

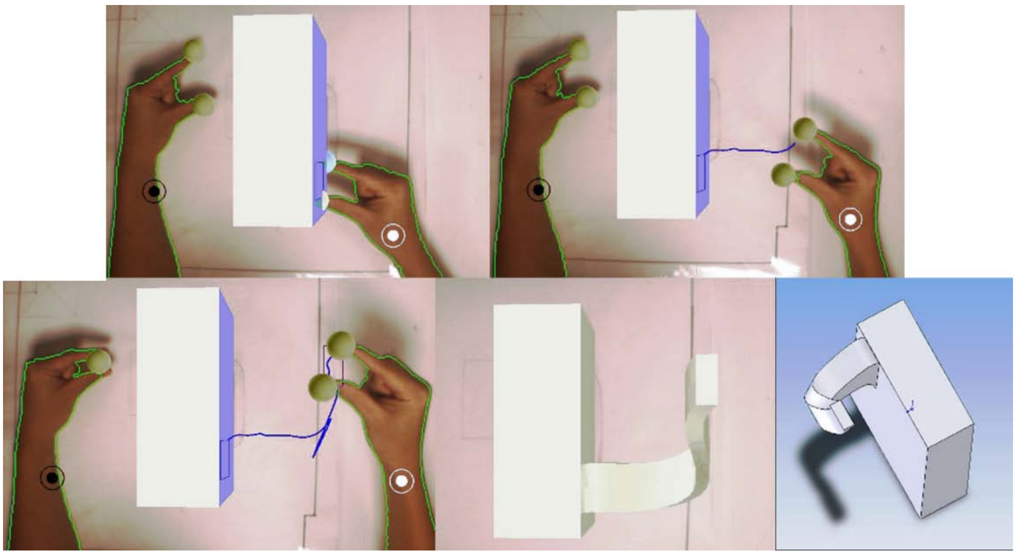


Figure 4.4: Freeform extrusion

(Top, from left to right: selection of face and definition of starting profile, definition of pulling path. Bottom, from left to right: definition of ending profile)

4.4.3 Editing

As the 3D models generated in SolidWorks are represented using Part-Feature Trees, and each part has a hierarchical tree of features associated with it, the user can edit a design in ARCADE by selecting the features to be edited using the pinching gestures. Depending on the types of feature selected, different editing operations can be performed.

The size, position and type of a "building block" feature can be modified. The user can modify the size by pinching one hand and moving his hand in the 3D design space; the new size will be the relative distance between the hand and the base of the feature. The position of a feature can be modified by dislodging and moving it to a new position. The feature type can be changed by selecting a new type.

An "extrusion" feature has more editing options, namely, the profile and the pulling path. The user has to select the parameters to be edited. The 2D shape and size of a profile can be edited. To change the shape of a profile, the user will have to select a new shape and define its size. To edit a pulling path, the user has to select the control points and amend their positions by moving them to new locations to obtain the desired path. These editing operations will only involve one hand and the editing process ends when the user's free hand pinches.

A feature can be deleted. The entire design can be edited by scaling it to a desired size using the “Scale” function. An “un-do” function is provided for the user to correct any wrong actions.

4.4.4 Building Block versus Extrusion

The building block approach is considered to be more intuitive than the extrusion approach as most users are more familiar with the building block concept compared to the extrusion from a 2D profile concept. As a result, the building block approach is the main modeling operation used to create 3D models in ARCADE, with the extrusion approach used as a supportive operation to add design features to the 3D model. In general, the 3D models are created using building blocks and more details can be added to via design features created using the extrusion approach.

4.5 Comparison with Conventional CAD System

The building block approach and the extrusion approach have been compared with a conventional CAD system using task analysis. The design task for comparison between the building block approach and CAD software is the creation of a 3D block. The design task for comparison between the extrusion approach and the CAD software is the extrusion of a feature. Table 4.3 shows the comparison results for the task analysis of the building block approach and CAD software while Table 4.4 shows the results for that of the extrusion approach and CAD software.

Table 4.3: Comparative task analysis between the building block approach and conventional CAD software in the creation of a 3D block

Building Block Approach	Conventional CAD Software
1. Select block primitive 2. Indicate start of creation by pinch gesture 3. Determine the size of the block (mental activity) 4. Move finger to define the size of the block 5. Confirm the completion of creation by pinch gesture	1. Select plane for 2D sketch 2. Select shape for 2D sketch 3. Determine the size of the 2D sketch (mental activity) 4. Sketch shape on plane by click-and-drag 5. Confirm the completion of 2D sketch 6. Select extrusion option 7. Determine the depth of extrusion (mental activity) 8. Enter the depth of extrusion 9. Confirm the completion of extrusion

From Table 4.3, it can be seen that the building block approach take fewer steps (five steps) to create the same 3D model as compared to the CAD software (nine steps). In addition, the mental activity of determining the size of the block is direct compared to CAD software, which requires the user to first determine the size of the 2D sketch followed by the depth of the extrusion. This reduces the cognitive load on the user, as there is no need to break down the 3D dimensions to a 2D sketch and a depth in order to create the 3D block. The direct mapping of the 3D interaction and operation with the 3D spatiality of the 3D model makes the building block approach more intuitive than the conventional CAD software.

Table 4.4: Comparative task analysis between the extrusion approach and conventional CAD software in the normal extrusion of a feature

Extrusion Approach	Conventional CAD Software
1. Select surface for extrusion 2. Define the position of the profile by pinch selection of the point 3. Define the type of the profile 4. Determine the size of the extruded feature (mental activity) 5. Move thumb and finger to define the size of the profile 6. Move the hand to define the extrusion path 7. Confirm the completion of extrusion by pinch gesture	1. Select surface for 2D sketch 2. Select shape for 2D sketch 3. Determine the size of the 2D sketch (mental activity) 4. Sketch shape on plane by click-and-drag 5. Confirm the completion of 2D sketch 6. Select extrusion option 7. Determine the depth of extrusion 8. Enter the depth of extrusion 9. Confirm the completion of extrusion

The number of steps taken by the extrusion approach (seven steps) is fewer than the conventional CAD software (nine steps). As both approaches use the same method for creating the 3D model, there is little difference between the steps taken. The reduction in steps in the extrusion approach is due to the direct determination of the 3D dimensions of the extrusion feature and the ability of the extrusion approach to determine the 2D profile and path in a single operation compared to the conventional CAD software which uses two operations (2D sketch and extrusion). One advantage that the extrusion approach holds over the conventional CAD software is that the determination of the size of the profile and the extrusion path are performed by concurrent user actions. The user defines the size of the profile by varying the distance between the thumb and the index finger and the path is defined by the distance between the index finger and the surface that the feature is extruded from. This advantage will be amplified when the extrusion becomes more complex, such as a multiple profile or freeform path extrusion.

From the task analysis, it can be concluded that the building block approach and the extrusion approach for creating 3D models in ARCADE is more intuitive than the conventional CAD software. Both approaches take fewer steps to obtain the same 3D model and uses direct mapping of 3D interaction and 3D modeling operations with the 3D spatial dimensions of the 3D model.

4.6 Designing with Real Objects

As ARCADE is an AR design system, real objects are used to create the 3D models as well. Real objects can be utilized in two ways for design generation. Firstly, they can be used as spatial references for the 3D models. The user can create the 3D model by using his hands to size up the real object. This will ensure that the 3D model will fit with the use environment and the user can contextualize the 3D model with the real object.

Secondly, they can be reconstructed as building blocks and tracked by attaching a marker to them. They can be added to the 3D models using the building block modeling approach and combined with the virtual 3D models to form the final design of the product. This will help to save the user from creating 3D models from scratch when there are already existing real objects that can be reconstructed and utilized for design generation in ARCADE.

5. Interactive Functional 3D Model using Function-Behavior-Structure Modeling

5.1 Introduction

3D models represent the geometry of a design visually. Like 2D sketches, there is not much interaction with them that can demonstrate how the product will work. In order to understand the workings of a product, physical mock-ups are built usually. The interactivity of the physical mock-ups makes them useful in sharing and demonstrating the design to others. However, it is time-consuming to build physical mock-ups for all the ideas that are generated during conceptual design and the physical mock-ups generally only demonstrate one aspect of the design.

Interactive 3D models are 3D models that are able to simulate the behavior of a product when the user interacts with it. In ARCADE, functional 3D models (F3DM) are created using a Function-Behavior-Structure Modeling framework in the FBSMM. The functions, behavior and product structure of the F3DM are reasoned so that the F3DM will behave like a real product when the user interacts with it. In addition, the F3DM incorporates a physics model, which will simulate the physical interactions it will have with the surrounding real objects.

In this chapter, the FBS modeling framework and the reasoning processes to generate the F3DM from the user's input of PUM and 3D model will be presented. The implementation of the physics simulation for F3DM will also be described.

5.2 Definition of termsThe definition of Function varies for different researchers (Chandrasekaran, 2005). It is difficult to define function and behavior independently. Function can be viewed as either Purpose Function or Action Function, and both types of functions exist in a product. Purpose function is a description of the designer's intent or the purpose of a design whereas action function is an abstraction of the intended and useful behavior that an artifact exhibits. In this research, only action function is considered and reasoned i.e., function refers to Action Function and it is defined as the input-output flow of the action transformation between objects (Deng, 2002). The difference between function and behavior is the notion of time.

Behavior represents the state transition of the objects when they are serving their functions. There are generally two types of behaviors, namely, Expected Behavior and Unexpected Behavior. The former is designed into a product and defined from the functions that are associated with it. The Unexpected Behavior is behavior that a product will exhibit because of its working structure and the side effects of the product performing certain functions.

The definition of product structure is more straightforward and it represents the objects and their geometrical relationships and physical interactions of a product.

For ARCADE, the approach is to add behavior simulations to the 3D models that are created so that the user can interact with the 3D models and modify them in the same design environment, which reduces the time taken between design iterations.

5.3 Functional 3D Model

A functional 3D model (F3DM) combines function models with 3D models. It allows the designers to consider the functional and geometrical aspects of a design concurrently and provide them with early functional prototypes for evaluation. Function models are abstract and do not represent the design geometrically. It is possible that a functional design could be functionally feasible but not physically feasible. 3D models only provide graphical information about the geometry of a design and the user cannot interact with them to understand the workings of the product.

A F3DM of a component contains the geometrical information in the form of a 3D model, and the functional information in the form of a basic function model. A product F3DM links the functional information of all the components that it contains and its 3D model consists of the 3D models of the components in the designed geometrical layout. The behavior of the product F3DM is derived from the functional and geometrical relationships of the components' F3DMs. Hence, a product F3DM can represent the functions and the functional relationships between components with a concrete representation of the product structure in the form of 3D models, which leads to a better representation of the behavior of a product.

In order to reason the function-behavior-structure (FBS) of a product, a multi-level FBS modeling framework using customized FBS primitives has been developed for the F3DM. Using this modeling framework, the functions of a product can be reasoned and modeled from a set of high-level functions, which is captured using a Product Use Model (PUM). The PUM represents the desired user-product interactions. A database of FBS primitives and FBS modeling rules are used to perform the FBS reasoning. When the reasoning process is completed, i.e., the functional linkages, geometrical relationships and behavior among the product's components have been established, the product will be represented as F3DMs.

5.4 Multi-level FBS Modeling framework

A multi-level FBS Modeling framework has been developed to create the functional 3D model from the user input of PUM and 3D models as described in the conceptual design methodology using ARCADE in Chapter 3. It contains three levels, namely, the top level of FBS modeling language (FBSML), the middle level of the archetype product model (APM), and the bottom level of the design candidates.

The top level FBSML is defined using Web Ontology Language (OWL) classes, which are built with different axioms that define them based on the relationships they have with other classes. The FBSML, which has hierarchical classes and a four-element FBS primitive class (Section 5.5), is created as the foundation for functional reasoning. Function chains can be formed using a combination of FBS primitives, and a function model of a product can be represented as a combination of FBS primitives and function chains (Section 5.7). The product structure model can be derived from the function model by rearranging the function chains and FBS primitives to consist of Objects-Pairs, which capture the functional relationships between two components (Section 5.8). The functional relationships can be reasoned to obtain a set of geometrical rules that must be satisfied for the components to fulfill their functions. The behavior model of the product class consists of the expected and unexpected behavior. Expected behavior is derived from the designed functions whereas unexpected behavior is inferred from the functions and product structure (Section 5.9). The product class contains the function model, product structure model and behavior model.

The middle level APM is an OWL class that is defined as an instance of the product class. It is a meta-model (Yoshioka et al., 2004) which contains the required components, required functions, behaviors and design rules of a product as defined by its function model, product structure model and behavior model. The APM of a product is an abstract representation of the product, which describes its functionalities and product structure conceptually and contains design rules that must be satisfied for the design to be feasible. The design rules are inferred from the functional and geometrical relationships between the components. In the OWL language, they are generated as the prerequisite conditions for the APM class. Different design concepts are represented as different APMs and design variants can be built based on existing APMs.

For example, for a car APM, wheels, chassis, drivetrain, engine and steering wheel are a few of the necessary components, “to move when accelerated” and ‘to change direction when steered” are some of its required functions, and “four wheels must be aligned” is one of its design constraints. A sport car APM will inherit the car APM with additional components and functions, such as sports tires and spoiler. The APM of a product and its design rules are generated automatically from the function, product structure and behavior reasoning processes performed on defining the product class after the user has defined the PUM and 3D models of a product.

The bottom level of the design candidate is an instance of the APM and it inherits the functions, required components, behavior and design rules. In addition, it contains the design parameters that define a specific design of a product from the user-generated 3D models. It consists of the geometrical relationships between components, the dimensions of the components, the assembly configurations and the functional specifications. Design verification and evaluation can be performed on various design candidates created by reasoning the design candidates with the design rules of the APM.

The product F3DM is created for each unique design candidate and contains information from the Product Class, the APM and the design candidate to represent the functions, behavior and geometry of a design. In addition, it contains information on the corrections that are required of the design candidate so that it is functionally feasible as defined by the APM. In order to simulate product behavior, the behavior model is referenced to create the necessary simulations.

The simulations will be fed back to the user when the user interacts with the functional prototype of the product in the AR environment. The product F3DM represents the functions of the product conceptually and geometrically, contains information for checking the feasibility of the corresponding geometrical design, and provides a mechanism for simulating the functional behavior of the product. This allows the designer to have a better understanding of the inter-relationships of the functions, behavior and structure of the product and provides them with a working AR prototype for testing during conceptual design. Figure 5.1 shows the multilevel FBS modeling framework.

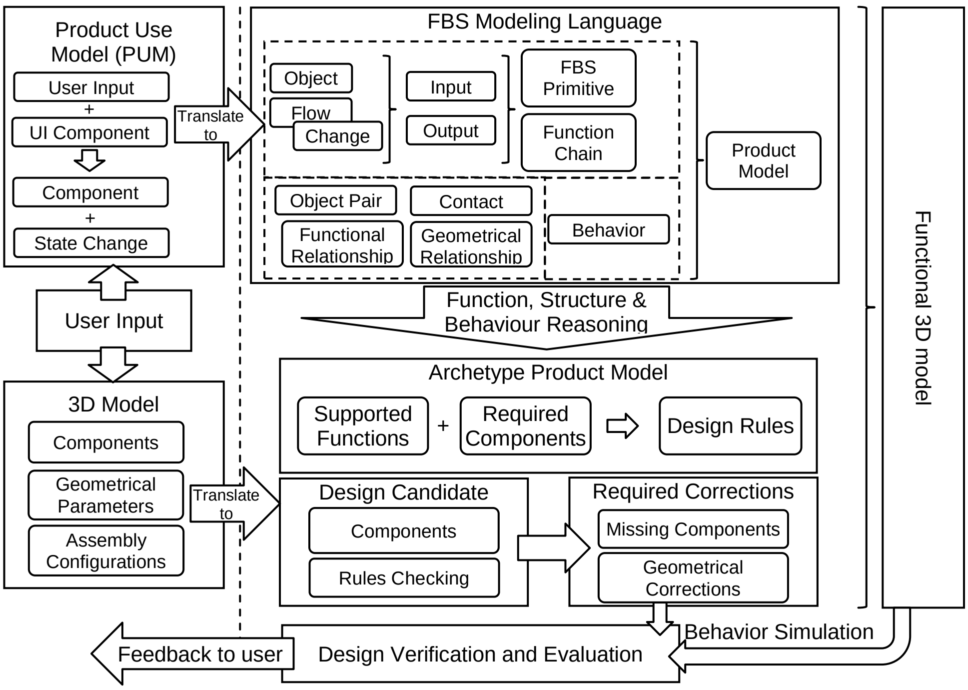


Figure 5.1: Multi-level FBS Modeling Framework developed in ARCADE

5.5 Function-Behavior-Structure Modeling Language

The FBSML is similar to some of the function modeling languages used in systems, such as KIEF (Yoshioka et al., 2004) and Schemebuilder (Bracewell & Sharpe, 1996). It uses the device-ontology approach of representing function as function-block diagrams. The product structure is represented using Object_Pairs in FBSML. They are similar to bond graphs, which generally link two objects with flows, with additional linkages involving the geometrical relationships and contacts between objects. While it is possible to modify and implement other languages for the system, it is more practical to develop one from scratch that addresses the needs of the reasoning processes that are performed. The main features of FBSML include:

1. A four-element FBS_Primitive class is used to represent function and contains information of the input, output flows, and the objects involved. FBS_Primitives can be combined to form the Function_Chain class to represent the functional flow for more complicated functions.
2. An Object_Pair class that can be derived from the FBS_Primitive and Function_Chain classes and link two objects with information about the changes in flows, the contacts the objects are having and the geometrical relationships they must have.
3. The role of the user interaction with the product is accounted for in FBSML. The user inputs are represented as a type of flow in the Flow class. This enables the system to represent product behavior in the form of a Behavior class that contains information on the user input and the resultant behavior.

This section describes the classes and their relationships in the FBSML. For clarity, a FBSML class will be in **bold**, an FBSML instance will be in *italics* and the name of the FBSML instance will be in quotes, " ". There are two element classes, namely, **Flow** and **Object**. A *Flow* and an *Object* are required for **Input** and **Output**. **FBS_Primitive** will have only one *Input* and one *Output*. **Function_Chain** will have an *Input*, an *Output* and a *Body*, which contains an ordered list of *Input/Output*. **FBS_Primitive** and **Function_Chain** form the main composition of the function model. For the structure model, it is mainly made up of **Object_Pairs**. An **Object_Pair** is a rearrangement of the **Function_Chains** and **FBS_Primitives** in an object-oriented manner and it has two *Objects*, a few *Functional_Relationship*, where each consists of two *Flows*, a few *Contacts* and *Geometrical_Relationship*. A **Functional_Relationship** contains two *Flows* that pass through the two *Objects* in an *Object_Pair* as derived from its original *FBS_Primitive*. The **Contact** contains information on the way the two *Objects* are connected. The **Geometrical_Relationship** is the geometrical relationship that the *Object_Pair* will have. **Behavior** is used to represent the behavior that the product will have when the user interacts with it. *Behavior* has a single *Input* to represent the user interaction and a list of *Outputs* as the behavior associated with the Input. **Product** has a few *FBS_Primitives* and *Function_Chains* as its function model, inherits the *Object_Pairs* to form its structure model and has different *Behavior* to form its behavior model. This representation of **Product** provides the foundation to understand its function, behavior and structure. The functions are in the form of *FBS_Primitive* and *Function_Chain*, the behaviors are derived from

Outputs of the *Function_Chains* and related *Objects* and the structure consists of the *Object_Pairs* and their *Functional_Relationships*, *Contacts*, and *Geometrical_Relationship*. Figure 5.2 shows the ontology graph for the various classes in the FBSML. The various classes are explained in details in the following paragraphs using a hair dryer as an example product.

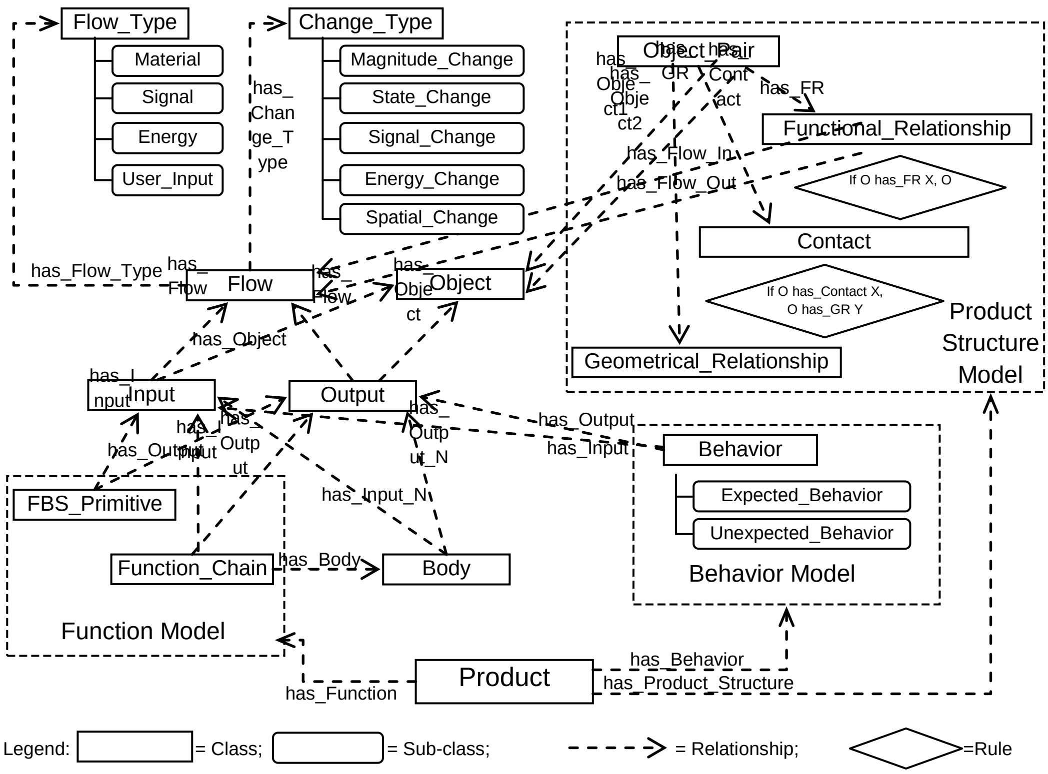


Figure 5.2: Ontology graph of the various classes in the FBSML

5.5.1 Flow Class

Function modeling in FBSML represents functions as the flow transformation between inputs and outputs. **Flow** can be broken down into the **Flow_Type** and **Change_Type**. **Flow_Type** represents the material, signal and energy flows that have been introduce (Pahl et al., 2007). In addition, the user inputs from the PUM are included as a special kind of Flow_Type that are generally used for **Input**. For a hair dryer, the material Flow_Type will consist of the air, the energy Flow_Type will consist of the electrical energy of the power supply, thermal energy of the heating unit, and mechanical energy of the fan, and the user input Flow_Type of pressing the button to turn on the hair dryer and adjust the speed of hot air. Some examples of user inputs are “Press”, “Turn”, “Pull”, “User_Move”, and “Insert”.

It is not sufficient to use only **Flow_Type** as the input and output flow, e.g., changes in the magnitude of a flow cannot be captured, certain physical phenomena do not involve a change in the flow such as a function to hold a position and a function to maintain speed. Therefore, **Change_Type** is created to capture the changes in flow. There are five types of changes, namely, **Magnitude_Change**, **State_Change**, **Signal_Change, Energy_Change** and **Spatial_Change**. **Magnitude_Change** changes the magnitude of the flow and can be applied to all three types of flows. Under **Magnitude_Change**, a flow can be added, removed, increased and decreased. **State_Change** is applicable to only material flow, which undergoes a change in state, for example from solid to liquid or from cold to hot. **Signal_Change** is applicable to signal flows whereby the signal is converted to other forms of signal or information. **Energy_Change** is applicable to energy flow where a form of energy is converted into another form. **Spatial_Change** is used to describe spatial changes, such as static, movement and storage for mainly material flows. Most *Spatial_Changes* are preceded by *Energy_Changes* or *Magnitude_Changes* for their inputs except *Storage*, which is used to represent materials that are stored in a certain object. Each *Flow* has a *Flow_Type* and a few *Change_Types* depending on the transformation to represent the input to or output flow from an *Object*. User input *Flow_Types* generally will not have any *Changes_Types* except in cases where the user has to apply varying magnitudes of forces on the product.

Using the hair dryer as an example, the speed of hot air from a hair dryer undergoes magnitude changes as the user adjusts the settings, "AirTemperatureIncreaseMoving" ⇒ "AirTemperatureIncreaseSpeedIncrease Moving". The air going through the hair dryer changes state from normal to hot air, "Air" ⇒ "AirTemperatureIncrease". The heating unit in the hair dryer converts the electrical energy from the power supply to thermal energy "ElectricalEnergyECToThermal" ⇒ "ThermalEnergyECfromElectrical". Figure 5.3 shows the flow transformation from "AirTemperatureIncreaseMoving" to "AirTemperatureIncreaseSpeedIncreaseMoving" and their *FlowType* and *ChangeTypes*.

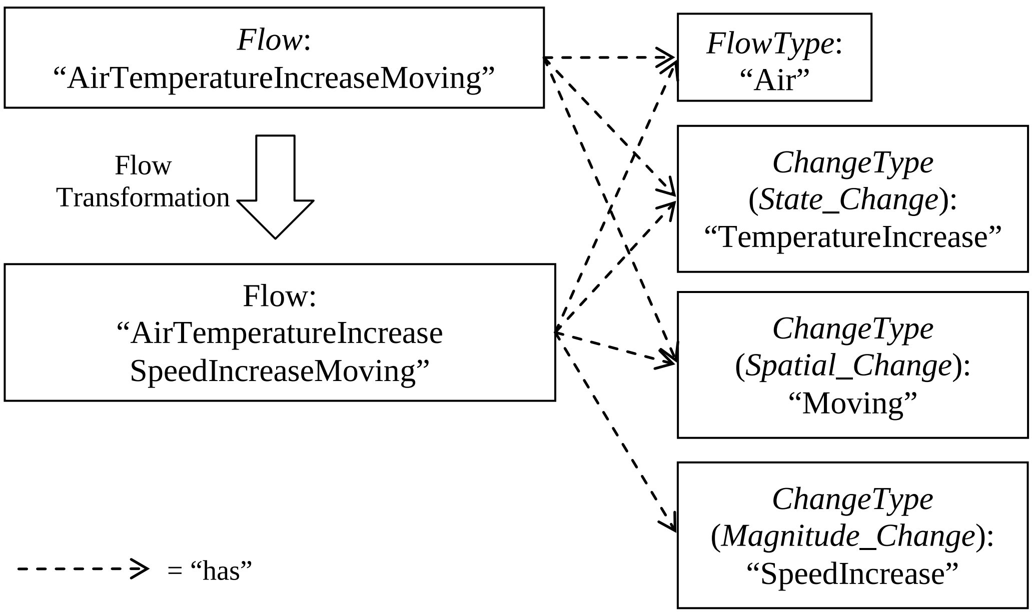


Figure 5.3: Flow Transformation representing the change in moving air speed for the hair dryer example

5.5.2 Object Class

Object represents the physical entity, artifact, and component. For **Input**, **Output** and **Object_Pair** classes, *Object* is the physical entity that is involved. An *Object* can be a **Flow_Type** in situations where material flow transformation occurs. The product structure is expressed by *Object_Pairs* and is derived from the list of objects that a product has and the functional relationships defined by its *FBS_Primitives* and *Function_Chains*. A hair dryer will have a power supply, a heating unit, a fan and a switch as some of its *Objects*.

5.5.3 Input and Output Classes

Input and **Output** represent the input and output flows and object transformations of a function. Therefore, they consist of a **Flow** and an **Object**. For an *Input*, the *Flow* represents the incoming change in material, signal, energy and user input and the *Object* represents the object that brings this change. For an *Output*, the *Flow* represents the resulting change in material, signal and energy, and the *Object* represents the object that undergoes this change. A naming convention is used to differentiate *Input* and *Output* with the same *Flow* and *Object*, with *Input* having the name of the *Flow* preceding that of the *Object* and *Output* having the opposite. This also means that the *Input* of one *FBS_Primitive* can be the *Output* of another *FBS_Primitive* and this is the underlying principle in the formation of **Function_Chains** from **FBS_Primitives**.

For a hair dryer, an *Input* will be “Press-Switch” with “Press” as the *Flow* and “Switch” as the *Object*, which represents the need for the user to press the switch in order to interact with the hair dryer. The corresponding *Output* could be "HairDryer-AirTemperatureIncreaseMoving" with "HairDryer" as the *Object*, "AirTemperatureIncreaseMoving" as the *Flow* consisting of "Air" as the *Flow_Type* and "TemperatureIncrease" (*State_Change*) and "Moving" (*Spatial_Change*) as the *Change_Type*.

5.5.4 FBS Primitive Class

FBS_Primitive is the basic function unit in FBSML and consists of an **Input** and an **Output**. A valid *FBS Primitive* is one which has different *Inputs* and *Outputs* as there is no flow or object transformation when there is no change in the *Flow* and *Object* of the *Input* and *Output*. An **FBS_Primitive** can have a change in **Flow**, **Object** or both between the *Input* and *Output*. Different types of functions in Hirtz et al.'s taxonomy (Hirtz et al., 2000) of functions can be represented using **FBS_Primitives** by modifying the changes in **Flow** and **Object** between the *Input* and *Output*. In addition to the flow and object transformation, structural information of the **FBS_Primitive** is captured using the assumption that transformation can occur only when the objects are connected. Therefore, different objects have to be connected via a *Contact* which is derived from structure reasoning.

An example of a *FBS_Primitive* in the hair dryer example will be "Torque-Motor-FanBlade-Torque" with the *Input* of "Torque-Motor" representing the input of the motor providing torque and the *Output* of "FanBlade-Torque" which represents the "FanBlade" obtaining "Torque" as a result of the *Input*. This FBS_Primitive also indicates that the "Motor" and "FanBlade" are connected and from structure reasoning as described in Section 5.8, they will share a "Mechanical_Contact_Coaxial" *Contact*. Figure 5.4 provides a graphical representation of the "Torque-Motor-FanBlade-Torque" *FBS_Primitive*.

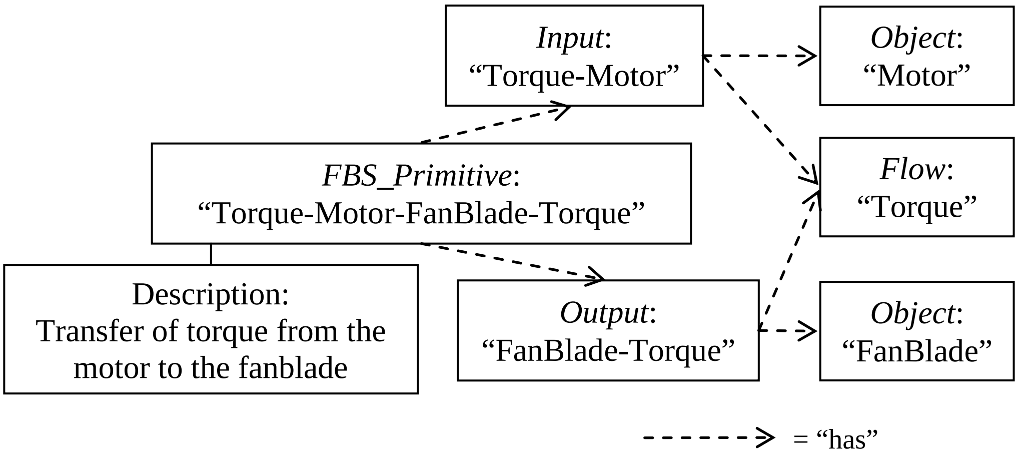


Figure 5.4: "Torque-Motor-FanBlade-Torque" *FBS_Primitive*

5.5.5 Function Chain Class

A **Function_Chain** is a sequential chain of FBS Primitives and consists of an **Input**, an **Output** and a **Body**. It is based on the causal (de)composition of functions with the assumption that *Output* of a preceding *FBS_Primitive* is the *Input* of the succeeding *FBS_Primitive*. As the connection between *Input* and *Output* consists of the same *Flow* and *Object*, the order of the *FBS_Primitives* can be captured using a *Body*, which stores the connecting *Input/Output* in an ordered list. The structure of the **Function_Chain** allows the function reasoning and (de)composition of the functions of a product using a simple recursive algorithm that searches for matching *Inputs* and *Outputs* from a database of *FBS_Primitives* and connect them to form a *Function_Chain*. The function model of the product is a composition of the *FBS_Primitives* and *Function_Chains* that satisfies the desired functions. A *Function_Chain* of the hair dryer can have the *Input* of "Press-Switch" and *Output* of "HairDryer-AirHotMoving" and the *Body* of the chain can be established using function reasoning.

5.5.6 Object Pair Class

Object_Pair is a rearrangement of the *FBS_Primitives* and *Function_Chains* that are reasoned for a product and is used specifically to represent the structure of the product. An *Object_Pair* consists of two *Objects* (*Object1* and *Object2*) that are connected by either the *FBS_Primitives* or *Function_Chains*. Each *Object_Pair* is unique and contains the *Functional_Relationships, Contacts* and *Geometrical_Relationships* of these two *Objects*. Using **Object_Pair**, the product can be analyzed in an object-oriented manner which bridges the gap between the abstract functions and the concrete geometries. From the *FBS_Primitive* of "Torque-Motor-FanBlade-Torque" (see Figure 5.5), an *Object_Pair* of "Motor-FanBlade" can be formed with "Motor" as *Object1* and "FanBlade" as *Object2*.

5.5.7 Functional Relationship Class

Functional_Relationship represents the functional relationships between two objects in the form of flows. Each *Functional_Relationship* has two flows, namely, *Flow_In* which represents the input *Flow* from *Object1* and *Flow_Out* which represents the output *Flow* to *Object2*. An *Object_Pair* can have different *Functional_Relationships* as defined by the function model of the product. The *Functional_Relationship* of "Motor-FanBlade" will be "Torque-Torque" with

"Torque" as the *Flow_In* and "Torque" as the *Flow_Out* and this can be inferred as the transfer of torque from the motor to the fan blade.

5.5.8 Contact Class

Contact represents the way the *Objects* in an *Object_Pair* are physically connected. Some of the supported *Contact* are "Mechanical_Contact_Coaxial", "Electrical_Contact", "Thermal_Contact", "Mechanical_Contact_Rigid_Joint", "Mechanical_Contact_1DOF_Joint", etc. The types of *Contact* are derived from the *Functional_Relationships* of an *Object_Pair* using a set of rules that infers the *Contacts* from the *Functional_Relationships*. For example, *Electrical_Contact* is derived from *Functional_Relationships* that contain at least one *Flow* that has *Electricity* as its *Flow_Type*. For "Motor-FanBlade", the two objects will share a "Mechanical_Contact_Coaxial" and "Mechanical_Contact_Rigid_Joint" as there is a transfer of torque for the motor to the fan blade and the axle of the motor and the fan blade will turn together.

5.5.9 Geometrical Relationship Class

Geometrical_Relationship represents the geometrical relationships that the *Objects* in an **Object_Pairs** must have due to their *Contact* and *Functional_Relationships*. They are derived from *Contact* that are mechanical mainly and can be used to derive the geometrical design rules that must be followed by the 3D models that are generated by the user. For example, the "Motor" and the "FanBlade" have a "Mechanical_Contact_Coaxial" and this leads to them having a "Geometrical_Relationship_Coaxial". Consequently, this leads to a geometrical design rule that defines that the 3D models of the "Motor" and "Fan Blade" must share a common rotational axis. Figure 5.5 shows the conversion of the "Torque-Motor-FanBlade-Torque" *FBS_Primitive* to the "Motor-FanBlade" *Object_Pair* and its relationships.

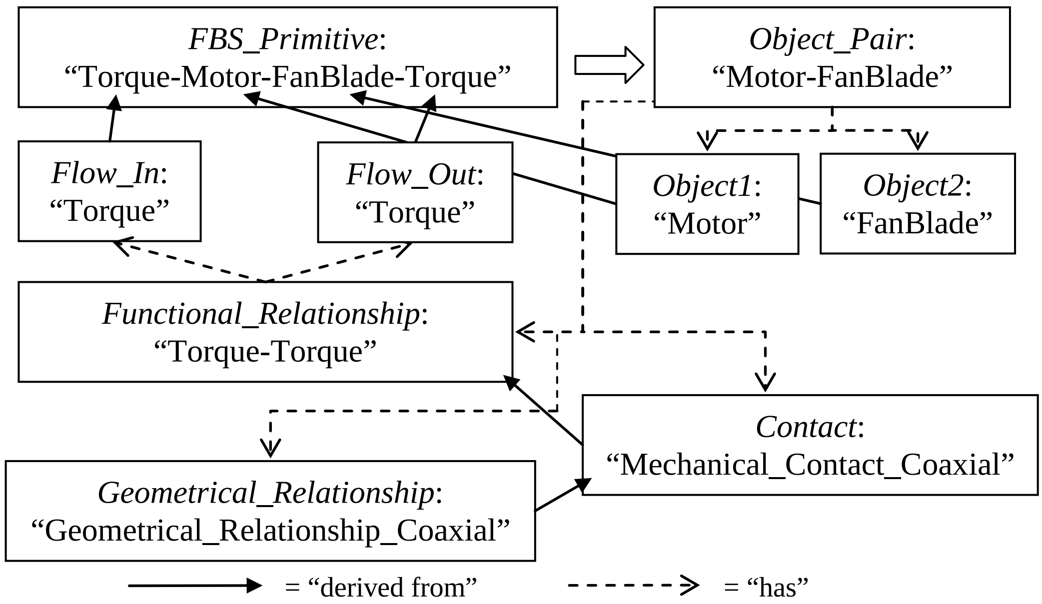


Figure 5.5: Conversion of the "Torque-Motor-FanBlade-Torque" *FBS_Primitive* to the "Motor-FanBlade" *Object_Pair* and its relationships

5.5.10 Behavior Class

Behavior represents the behavior that a product has when a user interact with it. The user interaction is represented by the *Input* of a *Function_Chain* and the series of *Outputs* are the associated *Expected_Behaviors*. The Unexpected Behaviors are reasoned from the *FBS_Primitives* database. In general, a *Behavior* will have a single *Input*, a few *Outputs* as the *Expected_Behavior*, stored in an "Expected_Behavior_List" and the *Unexpected_Behavior*, stored in an "Unexpected_Behavior_List". The **Behavior** is used for product behavior simulation whereby the product will behave according to the *Behavior* defined by its behavior model.

For the hair dryer example, it will have a Behavior, “Behavior-Press-Switch” consisting of the *Input* of “Press-Switch”. This will lead to the *Output* of hot air coming out from the hair dryer “HairDryer-AirTemperatureIncreaseMoving” (expected behavior) and other behavioral *Outputs*, such as the fan blade rotating “FanBlade-FanBladeRotate” (expected behavior), the heating unit heating up “HeatingUnit-HeatingUnitTemperatureIncrease” (expected behavior) and the sound made when the moving air pass the heating unit “HeatingUnit-AirMovingSound” (unexpected behavior). Figure 5.6 shows the behavior model of the hair dryer and the graphical simulation of the *Behavior*.

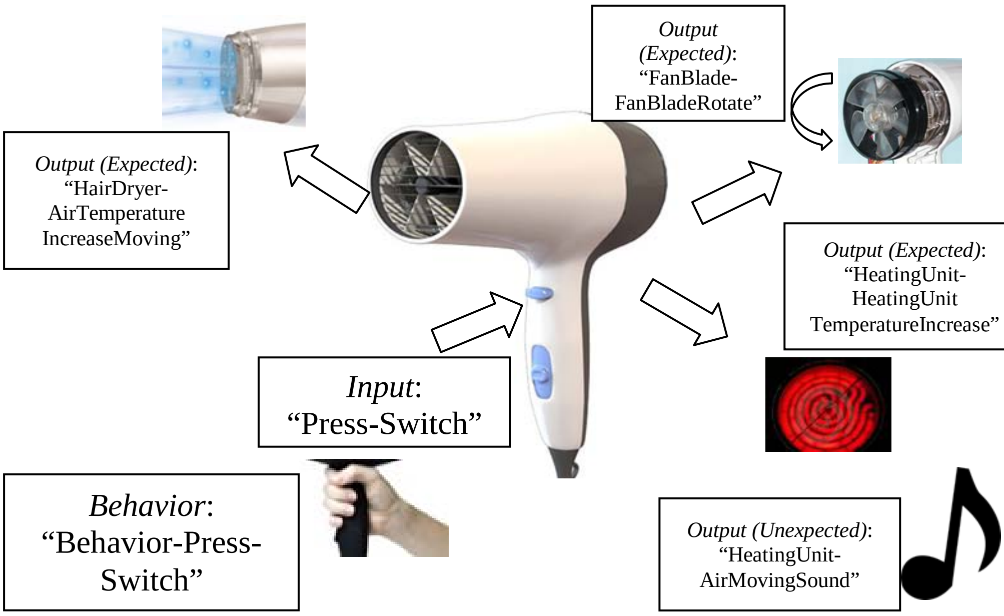


Figure 5.6: Behavior model of hair dryer example

5.5.11 Product Class

Product is used to represent the conceptual design of a product in the form of a function model, a structure model and a behavior model. The function model consists of the **Inputs** and **Outputs** that must be satisfied and functions which satisfy the *Inputs* and *Outputs*, in the form of **Function Chains** and **Function Primitives**. The structure model consists of the **Object_Pairs** which represent the product components. The behavior model contains the **Behavior** the product will have as a result of user interactions. A *Product,* which has its unique function, structure and behavior models, can be used to create an APM class, which will be used to reason the design candidates subsequently.

For example, a “Hair Dryer” *Product* can be created with the functions, structures and behavior as described in the preceding sections and this can be used to create a “Hair Dryer APM” APM class which is defined to have all the attributes of the “Hair Dryer” product. A design candidate of a hair dryer, “Hair Dryer DC1” is created as an instance of the “Hair Dryer APM”. The “Hair Dryer DC1” has the geometrical parameters of its components, in the form of the 3D models created, and these parameters can be reasoned and checked against the design rules of the “Hair Dryer APM” to verify the design. In addition, the 3D models of the components will be reasoned for their behavior when the “Hair Dryer DC1” is tested as a functional prototype.

5.6 Database and Data Extraction

Fifty household consumer products have been studied to extract their FBS primitives. They include appliances, such as vacuum cleaner, washing machine, and coffee maker, electrical appliances, such as television, radio, and personal computer, and common items, such as clock, chair and pen. Figure 5.7 shows some of the products used to build the database. Information of these products is taken from HowStuffWorks website (HowStuffWorks, 2013). 253 *FBS_Primitives* have been identified and stored in the database. A few of the common *FBS_Primitives* include "Press-Switch-Power Supply-Electricity", which describes the turning on of a device by pressing a switch to turn on the device and is found in all electrical devices, and Electricity-Power Supply-Motor-Torque, which describes the supply of electricity to a motor to provide torque by rotating and is found in most electrical devices with moving parts.

Figure 5.7: Some appliances and products used to create the database

The process of extracting the *FBS_Primitives* begins by establishing the usage of the product in the form of PUM. Each set of User Input and Product Response of a PUM is then decomposed into sub-functions to form a *Function_Chain*. Each link of the *Function_Chain* is then extracted as a *FBS_Primitive*.

There are two type of decomposition: task and causal. Task decomposition is done when an *Input* or *Output* can be broken down into two independent sub-functions, which can serve their functions without each other and are combined together to achieve a new input or output. An example will be the combination of the heating unit providing heat to increase the temperature of the air surrounding it, "HeatingUnit-AirTemperatureIncrease", and the fan blowing the air, "Fan-AirMoving", for the hair dryer to achieve the *Output* of providing moving hot air, "Fan-AirTemperatureIncreaseMoving". When an *Input* or *Output* undergoes task decomposition, a task decomposition rule involving the *Input* or *Output* and the sub *Input* and *Output* that constitute it is recorded so that it can be applied for function reasoning (Section 5.7). The *Input* or *Output* will also be identified as a task-input or task-output.

Causal decomposition is the sequential ordering of sub-functions so that the final function can be achieved. An example will be the motor providing torque, which is then transferred to the fan blade so that the fan can achieve the function of blowing of the air around it. After extracting the *FBS_Primitives* and forming the *Function_Chains* of each product using both task and causal decomposition, the *FBS_Primitives* are analyzed to check if they will lead to additional *FBS_Primitives* that are side effects of them achieving the functions. For example, when the heating unit heat ups, the heating coil will also light up. Therefore, an additional *FBS_Primitive* of "Electricity-PowerSupply-HeatingUnit-Light" is found onto of the function *FBS_Primitive* of “Electricity-PowerSupply-HeatingUnit-Heat”.

After finding all the *FBS_Primitives*, each *FBS_Primitive* is studied for the mechanical contact between the objects that satisfy the functions. If a contact, which is necessary for the *FBS_Primitive* to fulfill its function, is identified between the two objects, the *Input Flow* and *Output Flow* are used to define rules that use the *Functional_Relationships* to determine the Contacts for the *FBS_Primitives* and *Object_Pairs*. For all the Contacts that have been identified, they are analyzed to determine the geometrical relationships that define such type of *Contacts* and *Contacts-Geometrical_Relationships* rules are formed from them. This process continues with establishing the constraints on the actual geometrical parameters that will follow these extracted *Geometrical_Relationships*. As such, geometrical design rules can be established from the *Geometrical_Relationships* that a pair of object must have in order to serve its functions.

From the database of *FBS_Primitives*, the *Output* of each *FBS_Primitives* is studied to check if it is possible to create visual simulation of it. The simulations that can be supported are limited to movements, quantity changes, size changes and heat in the form of changing colors. The *Outputs* that can be simulated are recorded together with the simulations that are used for it, and stored in a lookup table. This lookup table, which matches the supported *Outputs* with their corresponding simulations, will be used to determine the simulations that will be feedback to the user when the user interacts with the functional prototype.

5.7 Function Reasoning

The purpose of function reasoning is to understand the functionalities of a product and establish a working system that can perform the required functions. Function reasoning consists of breaking down high-level functions into sub-functions, which eventually can be linked to a physical structure or phenomenon which can fulfill these sub-functions. High-level functions can be defined as functions that must be satisfied by the product in order to meet its design requirements. In this research, the PUM defined by the user is the source of high-level functions. When the user has created the PUM, a list of *Inputs* and *Outputs* will be extracted from this PUM. These are the initial *Inputs* and *Outputs* to be reasoned by the system using function decomposition.

Function decomposition can be divided into two categories, namely, (i) task and (ii) causal decomposition. Task decomposition is used to decompose functions that can only be fulfilled with the combination of two or more independent sub-functions. A pair of functions is independent if each function does not require the other function in order to be executed. Causal decomposition is used to decompose functions into sub-functions that are dependent and hierarchical. A sub-function cannot be executed if its preceding function has not been executed. After function decomposition, a product will be represented as a combination of *Function_Chains* and *FBS_Primitives*. The structure model of the product is next established based on its *Function_Chains* and *FBS_Primitives* in the form of the *Object_Pairs*.

Figure 5.8 shows the function reasoning process. The description of the steps is as follow:

Step 1: Extract the initial *Inputs* and *Outputs* from the PUM

The PUM defines the desired functions of the product, and the *Inputs* and *Outputs* are extracted by taking the user inputs and interactions as the *Inputs* and the product response as the *Outputs*.

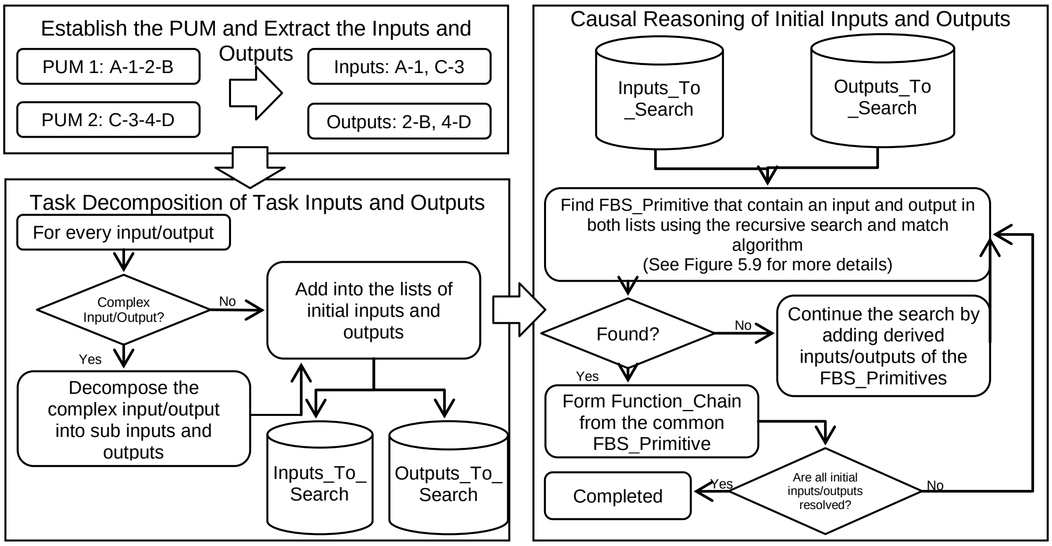


Figure 5.8: Function reasoning process

Step 2: Task Decomposition of Task Inputs and Outputs

The Inputs and Outputs are checked to determine whether they are task Inputs and Outputs as identified during data extraction. Task decomposition is performed using rules that break down the task *Input* and *Output* into a combination of sub Inputs and Outputs. Equation (5.1) shows the general form of a task decomposition rule. A constraint of the rule is that for a task *Input/Output*, the number of sub *Input/Output* must be greater than the number of sub *Output/Input*.

After task decomposition, the lists of initial inputs and outputs for causal decomposition can be established.

$$\textit{If Product } \boldsymbol{X} \textit{ has Task Input/Output } \boldsymbol{Y} \rightarrow$$

$$\textit{Product } \boldsymbol{X} \textit{ has Inputs, } \boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\ldots\textit{, and Outputs, } \boldsymbol{E}+\boldsymbol{D}+\boldsymbol{F}+\ldots \qquad (5.1)$$

Step 3: Causal Reasoning of initial Inputs and Outputs

The initial *Inputs* and *Outputs* are used to generate *Function_Chains* with unknown *Body*, and causal reasoning is used to determine the *FBS_Primitives* that are linked sequentially to produce the *Function_Chains* and establish their *Bodies.* The *Input* and *Output* of an unresolved *Function_Chain* is assigned according to the task decomposition rules, as certain *Input* must be matched with certain *Output* in order for the *Function_Chain* to fulfill its function. This type of unresolved *Function_Chain* is named as Type 1 Unresolved *Function_Chains*. The remaining *Inputs* and *Outputs* will then be randomly assigned to form other unresolved *Function_Chains*, named as Type 2 Unresolved *Function_Chain*. Type 1 Unresolved *Function_Chains* will be reasoned first to establish their *Bodies* followed by Type 2. For each unresolved *Function_Chain*, a recursive search and match algorithm that searches all *FBS_Primitives* to form the *Body* is deployed. This will continue until all the initial *Inputs* and *Outputs* can be associated with reasoned *Function_Chains* that have established *Bodies* between their *Inputs* and *Outputs*.

The recursive search and match algorithm consists of three phases, namely, (1) find the *FBS_Primitives* that match the *Input* and *Output*, (2) establish the link from the *Input* to the *Output*, and (3) determine the order of the *Function_Chains*. Figure 5.9 illustrates the algorithm.

A recursive “Match_FBS_Primitives” function is the main function used and it takes in a list of “Inputs_To_Search”, a list of “Outputs_To_Search” and a list of “Matched_FBS_Primitives”, which is used to store the matched *FBS_Primitives*. The function will stop in three scenarios. The first scenario is when the list of “Inputs_To_Search” or the list of “Outputs_To_Search” contains nothing, which leads to an error, as there is nothing to search for. The second scenario is when both lists reaches 253, which are the maximum number of *FBS_Primitives* and this imply that there will be no solution for the unresolved *Function_Chain*. The third scenario occurs when a *FBS_Primitive* which matches an *Input* from the “Inputs_To_Search” list with an *Output* from the “Outputs_To_Search” list. The FBS_Primitive will be added to the “Matched_FBS_Primitives” list upon the termination of the function in this scenario.

If there is no termination of the function, the function will continue by recursively invoking itself after adding the *Inputs* or *Outputs* to the “Inputs_To_Search” list and “Outputs_To_Search” list respectively and perform the search and match on the new lists. The process of adding *Inputs* is to find the *FBS_Primitives* that share the same *Inputs* in the current “Inputs_To_Search” list, convert their *Outputs* to *Inputs* and add them to the list. Adding of *Outputs* to the

“Outputs_To_Search” list is the reverse. For each recursive call of the “Match_FBS_Primitives” function, only one of the “Inputs_To_Search” list and the “Outputs_To_Search” list is added with new *Inputs* or *Outputs* and the list, which has fewer items, will be added with new items.

Phase (1) begins with adding the initial *Input* and *Output* of the unresolved *Function_Chain* to the “Inputs_To_Search” and “Outputs_To_Search” lists. These two lists are compared to check whether there is any *FBS_Primitive* that has the same initial *Input* and *Output*. If there is, this implies that the *Function_Chain* can be represented as one *FBS_Primitive*, and phases (2) and (3) do not need to be executed. If there is no common *FBS_Primitive*, the algorithm proceeds to Phase (2).

Phase (2) begins with adding *Inputs* and *Outputs* of *FBS_Primitives* that share the same initial *Input* and *Output* that are currently in the “Inputs_To_Search” and “Outputs_To_Search” lists. The “Match_FBS_Primitive” recursive function will be invoked to find the first matching *FBS_Primitive*, which is important as the existence of it means that the *Function_Chain* can be resolved.

In the case where the Function_Chain cannot be resolved as the number of items in both “Inputs_To_Search” and “Outputs_To_Search” list reaches 253, the Function_Chain is marked as irresolvable and the reasoning process will skip Phase (3) and move on to the next unresolved *Function_Chain*. If there is only one irresolvable *Function_Chain*, a new *FBS_Primitive* can be created using the

Input and *Output*, and this will be marked as a *FBS_Primitive* that requires some innovation to achieve its function. If there are two or more irresolvable *Function_Chains*, their *Inputs* and *Outputs* are swapped and reasoning will be performed again until there is one or no irresolvable *Function_Chains*.

Phase (3) is performed to establish the *Body* of the *Function_Chain* as Phase (2) only determines whether a chain can be formed based on the *Input* and *Output*. Phase (3) begins with creating new lists of "Matched_Inputs_To_Search" and "Matched_Outputs_To_Search" from the *Output* and *Input* of the first matching *FBS_Primitives*. The original "Inputs_To_Search" list containing only the initial *Input* and the "Matched_Outputs_To_Search" list are used to invoke the "Match_FBS_Primitive" function. Subsequently, the "Matched_Inputs_To_Search" and the "Outputs_To_Search" lists containing only the initial Output are used to invoke the "Match_FBS_Primitive" function as well. Whenever a matching *FBS_Primitive* is found, the process of creating new lists of "Matched_Inputs_To_Search" and "Matched_Outputs_To_Search" and invoking the "Match_FBS_Primitve" function with the original "Outputs_To_Search" and "Inputs_To_Search" lists respectively is repeated until the "Matched_FBS_Primitives" list is populated with *FBS_Primitives* that are able to link the initial Input to the initial Output of the unresolved *Function_Chain*. This is followed by ordering the *FBS_Primitives* in the "Matched_FBS_Primitives" from the initial *Input* to the initial *Output*. At the end of this process, the unresolved *Function_Chain* is deemed to have been solved with an established *Body* that links its *Input* to its *Output*. At the end of the causal reasoning process, which establishes the *Bodies* of all the *Function_Chains* of the product, the function model of the product in the form of *Function_Chains* and *FBS_Primitives* is established.

Figure 5.10 shows an example of function reasoning of the hair dryer example with an initial Input of “Press-Switch” and Output of “HairDryer-AirTemperatureIncreaseMoving”. The function reasoning process begins with converting the PUM into *Input* and *Output*. The *Input* and *Output* will be checked if they can be task-decomposed. “Press-Switch” is not a task *Input* and is added to the “Inputs_To_Search” list. “HairDryer-AirTemperatureIncreaseMoving” is a task Output and is decomposed into “HeatingUnit-AirTemperatureIncrease” and “FanBlade-AirMoving”. Both Outputs are added to the “Outputs_To_Search” list and causal reasoning will begin by first finding a match between “Press-Switch” and “HeatingUnit-AirTemperatureIncrease” or “FanBlade-AirMoving” from the database. No match is found and since there is less item in the “Inputs_To_Search” list, “Press-Switch is used to find FBS_Primitive that shared the same Input. “Press-Button-PowerSupply-Electricity” is found from the database and its Output is converted into an Input, “Electricity-PowerSupply”. It is then added to the “Inputs_To_Search” list and the second iteration of finding a match begins. This will continue until a match has been found. In cases where a match cannot be found, a new *FBS_Primitive* will be created from the *Input* and *Output* and the user will be required to define it.

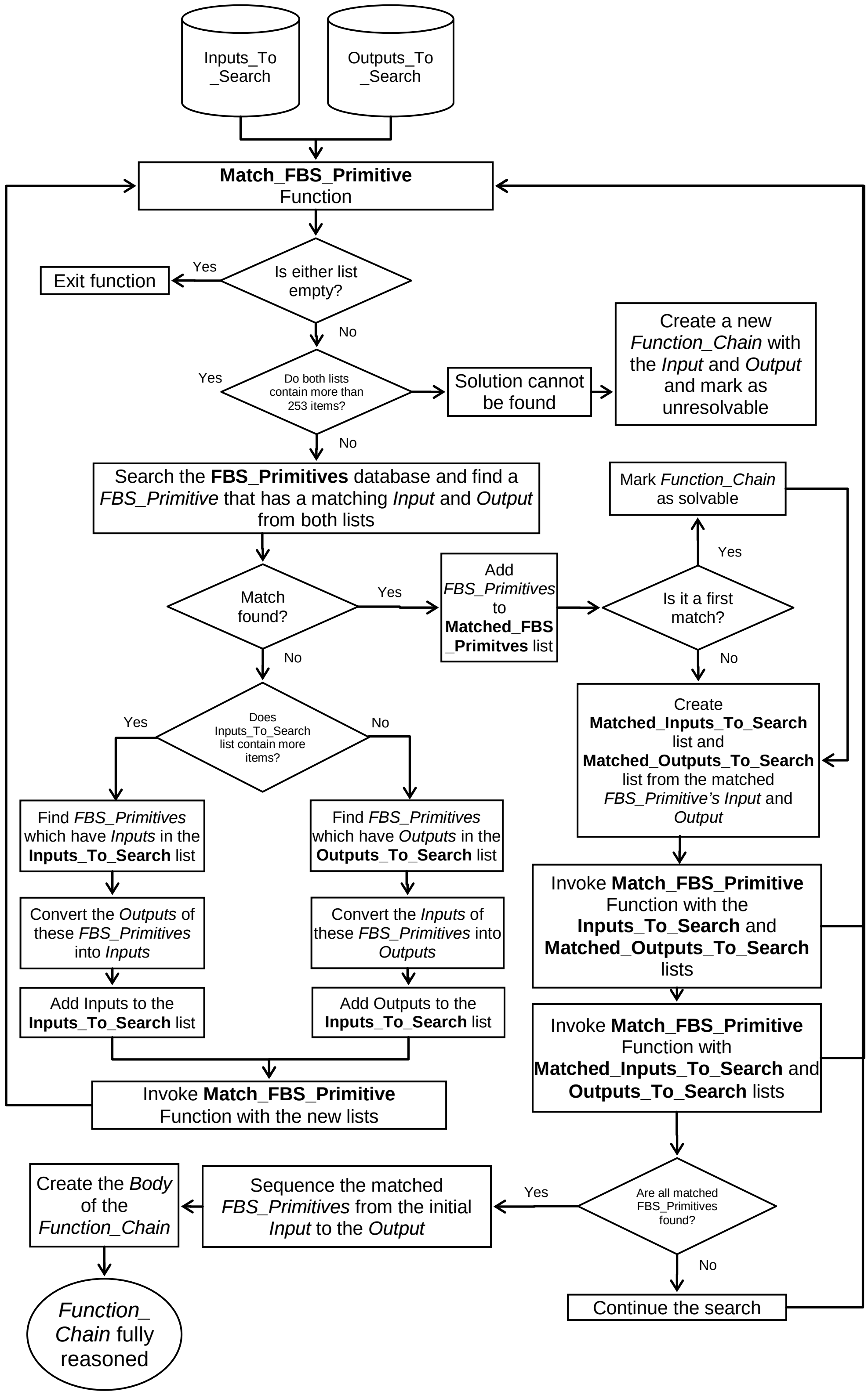


Figure 5.9: Recursive search and match algorithm used for causal reasoning

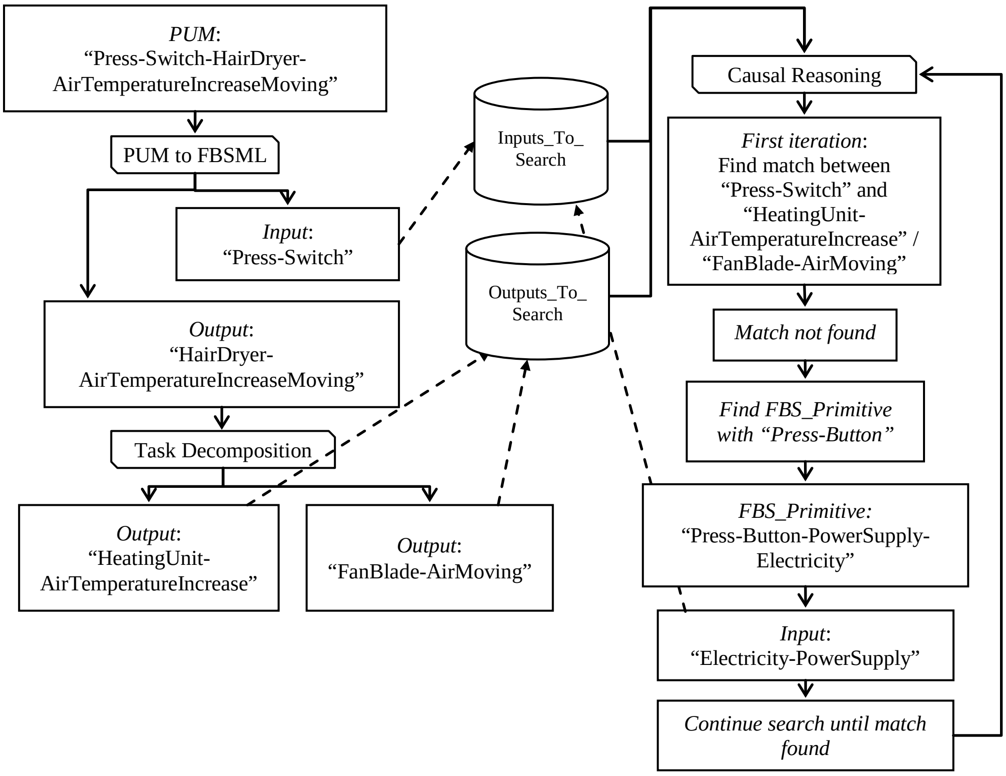


Figure 5.10: Function reasoning process for the hair dryer example

5.8 Structure Reasoning

Structure reasoning is performed after function reasoning to derive the product structure model from the functions. The product structure model consists of the organization of the components of a product to fulfill the functions. From the function model, the behaviors required can be established and by associating the behaviors to the physical features or phenomena, and the structure model is formed from these physical features and phenomena. Some examples of product structure reasoning are provided have been reported by Goel et al. (2009) and Umeda et al. (1996), which use function decomposition and qualitative analyses to establish the structure model of a product. One limitation of these approaches is that the structure does not consider the physical feasibility in terms of the geometrical relationships. In ARCADE, the product structure model is derived from the function model, and the functional relationships, contacts and geometrical relationships between the components can be established.

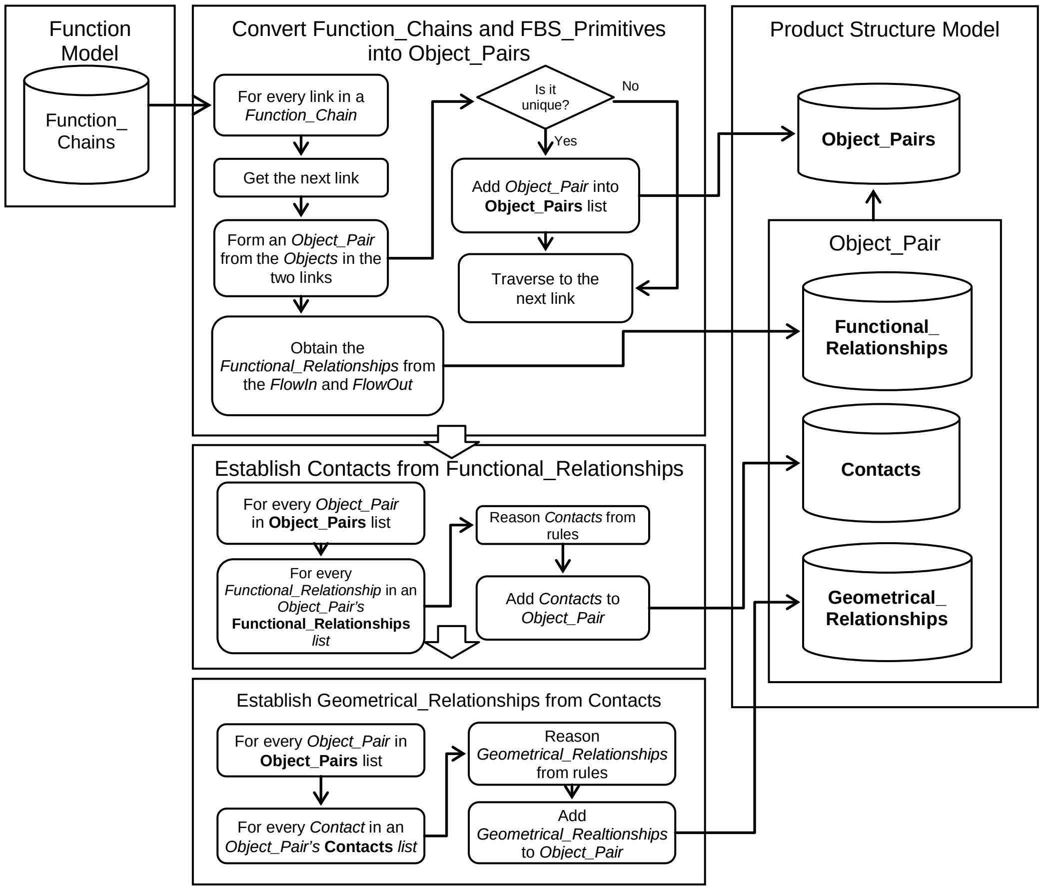


Figure 5.11: Structure reasoning process

Figure 5.11 shows the structure reasoning process. The description of the steps is as follow:

Step 1: Conversion of FBS_Primitives and Function_Chains to Object_Pairs

After function reasoning, the function model is established in the form of *Function_Chains* and *FBS_Primitives*. For each *Function_Chain*, the Objects in the first link (*Input*) and the next link are extracted as an *Object_Pair* and checked if it is unique. If it is, a new *Object_Pair* is created and stored in the product's list of *Object_Pairs*, "Object_Pairs_List". Otherwise, the existing *Object_Pair* will be referenced from the "Object_Pairs_List" for the next step of adding the *Functional_Relationships* of the *Object_Pair*. For each *Object_Pair*, the *Functional_Relationship* is derived from the *Flow_In* and *Flow_Out* of the two links of the *Function_Chain* and added to the *Object_Pair* if it has not been added. After the *Functional_Relationship* has been added to the *Object_Pair*, the next *Object_Pair* is extracted from the last link of the current *Object_Pair* and the next link. This process will continue until all the links in the *Function_Chains* and *FBS_Primitives* have been traversed.

Step 2: Establish Contacts from the Functional_Relationships for every Object_Pairs

The *Contacts* of an *Object_Pair* is derived from its *Functional_Relationships*. The rationale is that the Objects must be connected in a certain manner to satisfy the functions defined by their *Functional_Relationships*. Predefined rules, which are extracted from the database described in Section 4.1, are used to infer the *Contacts* from the *Functional_Relationships*. The general form of the rule is shown in Equation (5.2). The *Contacts* are added to the *Object_Pair* if they are not already present in this *Object_Pair*.

*If Object_Pair **XY** has Functional_Relationship **AB** →*

*Object_Pair **XY** has Contacts **C1, C2, C3**…* *(5.2)*

Step 3: Establish Geometrical_Relationship from the Contacts for every Object_Pairs

The *Geometrical_Relationship* of an *Object_Pair* is derived from its *Contacts*, in particular *Mechanical_Contacts*, as the geometry of the design is highly dependent on the mechanical connections of the components. Similar to step 2, predefined rules are used to infer the *Geometrical_Relationships* from the Contacts of an *Object_Pair*, and it is in the general form as shown in Equation 3. After obtaining the *Geometrical_Relationships* for each *Object_Pair* in the "Object_Pairs_List", the product structure model of a product is established and represented by its "Object_Pairs_List" .

*If Object_Pair **XY** has Contact **C1*** →

*Object_Pair **XY** has Geometrical_Relationship **GR1, GR2, GR3**…* (5.3)

An example of a structure reasoning process using a *FBS_Primitive* of "Torque-Motor-FanBlade-Torque" is: if *Object1* ("Motor") and *Object2* ("FanBlade") has a *Functional_Relationship* that contains *Flows* involving the direct transfer of torsion energy, "Torque-Torque", this implies that they have a "Mechanical_Contact_Coaxial". Since they have a "Mechanical_Contact _Coaxial", they are inferred to have a "Geometrical_Relationship_Coaxial" and "Geometrical_Relationship_Rigid_Joint". Figure 5.12 illustrates the structure reasoning process for this example.

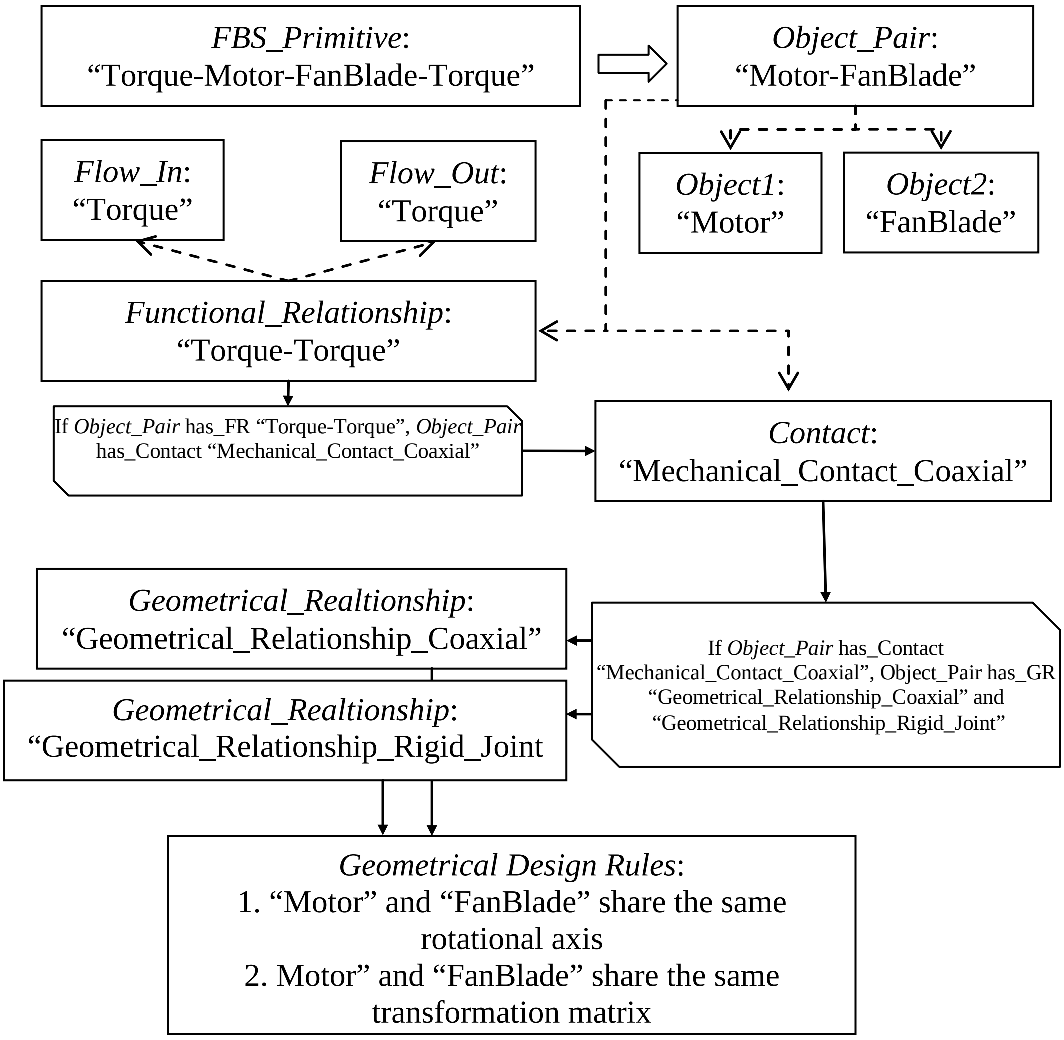


Figure 5.12: Structure reasoning for the "Torque-Motor-FanBlade-Torque" *FBS_Primitive*

5.9 Behavior Reasoning

The *Behavior* of a product is represented by the changes to the *Objects* and *Flows* and is in the same form as an *Output* consisting of an *Object* and a *Flow*. The purpose of behavior reasoning is to simulate product behavior as a result of user interactions. When a user interacts with the UICs, the system will reason the possible behavior. *FBS_Primitives* and *Function_Chains* are used to reason the Expected Behavior whereas Unexpected Behavior is derived by searching the

FBS_Primitives database that shared the same Input and Objects but different Flow_Out.

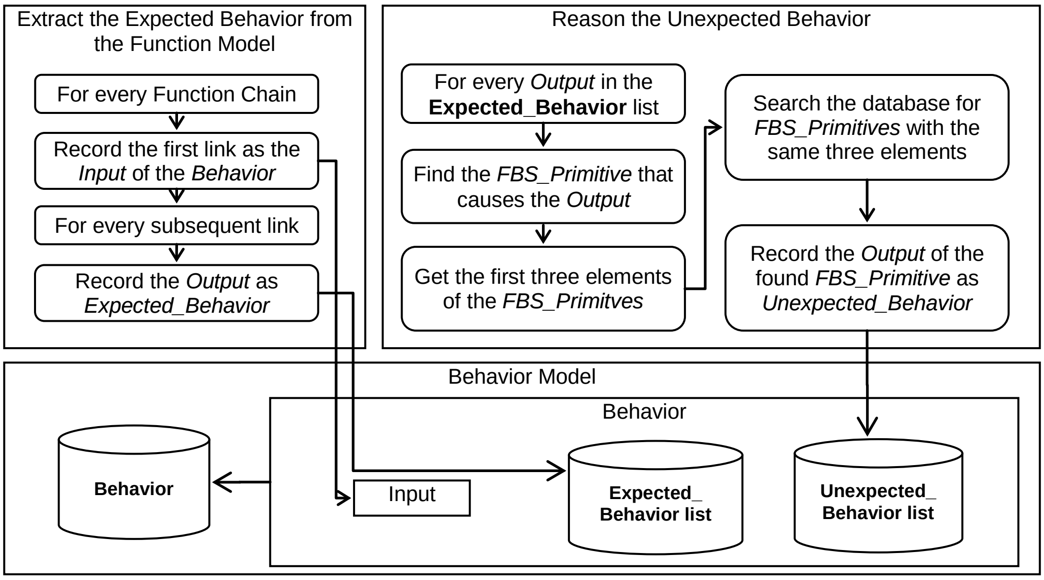


Figure 5.13: Behavior reasoning process

Figure 5.13 shows the behavior reasoning process. The description of the steps is as follow:

Step 1: Extract the Expected Behavior of the Product from the Function Model

The *Expected_Behavior* is designed for a product and can be extracted directly from the function model via the *Function_Chains* and *FBS_Primitives*. For each *Function_Chain*, the first link is recorded as the first *Input* and the *Output* for every subsequent link is recorded as the expected behavior and stored in a list of *Expected_Behavior*, "Expected_Behavior_List". An "Expected_Behavior_List" will be created for each *Function_Chain* in the same manner for all *Function_Chains* of the product.

Step 2: Reason the Unexpected Behavior of the Product

Unexpected_Behavior is the side effect from the *Expected_Behavior* of a product. In order to reason them, every *Behavior* in the "Expected_Behavior_List" created in Step 1 is reasoned to determine the *Unexpected_Behavior*. The reasoning process begins by (1) finding the *FBS_Primitive* that causes the *Behavior*, (2) taking the first three elements of this *FBS_Primitive*, i.e., the *Input* and the *Object* from the *Output* and (3) performing a search in the *FBS_Primitives* database for FBS_Primitives that have these three elements. The *Outputs* of the *FBS_Primitives* that have been found will be recorded as the *Unexpected_Behavior* of the product and be added to the "Unexpected_Behavior_List". After this reasoning process, every *Behavior* of the product will have an *Input*, an "Expected_Behavior_List" and an "Unexpected_Behavior_List" and the behavior model of the product is completed.

Using the example of the hair dryer, it will have a *Behavior*, "Behavior-Press-Switch" consisting of the *Input* of "Press-Switch". It can be reasoned that the "Expected_Behavior_List" consist of all the *Outputs* in the function model. Using the "AirMoving-HeatingUnit-HeatingUnit-AirTemperatureIncreaseMoving" *FBS_Primitive* as an example, the process for finding Unexpected_Behavior will use the first three elements, namely "AirMoving-HeatingUnit-HeatingUnit" as a basis to search the database for *Unexpected_Behavior*. A FBS_Primitive of

"AirMoving-HeatingUnit-HeatingUnit-AirMovingSound" has been found and this leads to the hair dryer having "HeatingUnit-AirMovingSound" as one of the *Outputs* in the "Unexpected_Behavior_List".

During product simulation, a product will behave according to the behavior model and the user interactions. For visualization of the behavioral simulation, only certain physical behaviors are simulated, e.g., representing a movement by changes of the objects' position and rotation, representing size changes by modifying the size of an object, and representing heat by changing the color of the object towards red for increasing heat and blue for decreasing heat. This is achieved by using a lookup table to link the simulations to the behavior and modifying the parameters of the 3D models to reflect the behavioral changes.

5.10 Overview of Reasoning Processes

Figure 5.14 provides an overview of the reasoning processes in ARCADE using an example of an electric toy car. The design process begins with the user defining a user interaction input of pushing (User Action) a Button (UIC) to obtain product behavior of the car (Object) moving (Response) in the PUM. This is parsed to the FBSML to obtain an initial input (Push-Button) and output (Car-Move) for function reasoning using the *FBS_Primitives* database. The function model obtained contains the *Function_Chains* and *FBS_Primitives* of the toy car and is used to define the supported functions in the APM. They can also be converted to form *Object_Pairs* to represent the product structure model, and reason the expected behavior. The required components of the APM are derived from the structure model. The geometrical design rules are defined from the *Geometrical_Relationships* of the *Object_Pairs*. The unexpected behavior is reasoned from the *FBS_Primitives* database. *Behavior* is next converted to the simulations in the lookup table to provide visualization of this behavior. From the APM, the 3D model of a design candidate is verified and evaluated to detect and correct possible issues with the design as described in Chapter 6. The revised design candidate is presented to the user as a functional AR prototype. When the system detects that the user have pushed the button on the car, a simulation of the car moving is presented to the user, i.e., changes in displacement of the car.

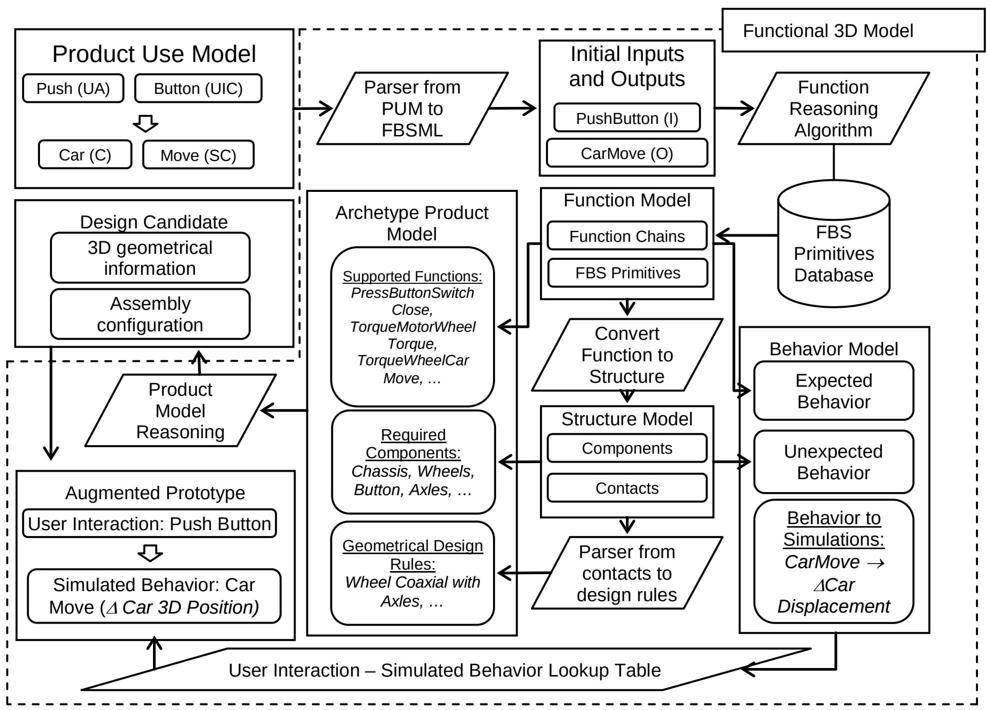


Figure 5.14: Overview of the reasoning processes in ARCADE

5.11 Physics Model

On top of simulating a product behavior when the user interacts with the F3DM, the dynamics of the F3DM can be simulated to provide more realistic and natural interactions between the 3D models and the real objects in the design space. For example, when the user switches on an electric toy car, the toy car will be simulated to move as a result of the FBS reasoning processes. If there is a real object that acts as an obstacle to the movement of the toy car, the physics model will simulate the dynamics of the collision between the moving toy car and the obstacle.

Simulation of rigid body dynamics is achieved with the implementation of ODE (Open Dynamics Engine) which uses a Boundary-Representation (B-Rep) physics model. Parameters, such as the weight and linkages of the components, form the physics model of the F3DM and are sent to ODE to compute the dynamics simulation. Real objects are tracked to obtain their spatial information for ODE to perform the dynamics calculation. When the virtual objects collide with both virtual and real objects, their positions, orientations and movements can be updated and simulated based on the rigid body dynamics simulation provided by ODE. The F3DM together with the physics model will be able to provide product simulations that behave similar to the final product when it is manufactured.

6. Design Simulation, Verification and Evaluation in ARCADE

6.1 Introduction

Design evaluation tools used for conceptual design typically use a scoring matrix to compare different solutions based on a certain set of criteria. This is an objective way of selecting the final concept for further development. However, these methods suffer from the following limitations:

- It is difficult to score the concepts on certain important criteria, such as technical feasibility and usability, due to the lack of quantitative information.
- The scores may lead to an average solution being selected over unique solutions that offer benefits that are not captured by the evaluation tools.
- The scores given for the solutions are generally subjective and research has to be conducted for each solution to be compared objectively. This is undesirable especially when there are many solutions.

In this research, the Design Simulation, Verification and Evaluation modules (DSVEM) in ARCADE aim to provide a more comprehensive analysis of the product. This is achieved in three ways:

1. The behavior of the product is simulated based on the F3DM generated using FBSMM. The F3DM behaves like the final product when the user interacts with it. This will allow the user to experience the usage of the product and evaluate its usability by actually using the product.

2. The form and function of the product can be verified using a design verification system that checks the 3D model created by the user against the function model defined using PUM. This will ensure the feasibility of the product and allow the user to understand the effects of the function on the geometrical aspects of the design.
3. Ergonomics of the design can be evaluated through the detection of hand strains when the user is handling the product. As the user's hands are tracked in ARCADE, algorithms are implemented to detect hand strain incidents that may occur when the user interacts with the product. This allows the user to evaluate the design based on the ergonomics without having to create a physical mock-up.

6.2 Behavioral Simulation of the F3DM

During the design evaluation process, the behavior of a product F3DM will be simulated when the user interacts with it. ARCADE will extract the input that triggers the *Behavior* from each of them in the behavior model created using FBSMM. The input is a user input from the PUM and contains a user action and UIC. When the user interacts with the product F3DM, the actions of the user will be tracked by the system.

Table 6.1 summarizes the supported user actions and their detection rules. If an action corresponds to that of a user input that triggers a *Behavior*, the system will check whether it is acting on the correct UIC. If it is correct, the behavior will be triggered. The simulations to be performed for the behavior are extracted from the

Behavior by first finding the *Outputs* in both the Expected and Unexpected Behavior Lists. Each *Output* is checked against the lookup table to see if it is supported. If it is, the physical parameters of the F3DM will be modified accordingly to realize the simulations. Table 6.2 shows the supported simulations and the corresponding modification to the physical parameters to realize them. Figure 6.1 illustrates the behavior simulation process during design evaluation.

Table 6.1: User actions and detection rules

User Actions	Detection Rules
Press	1. Fingertip in contact with the planar surface of the UIC 2. Fingertip moves in downward direction with respect to the planar surface of the UIC
Turn	1. Two fingertips in contact with the cylindrical surface of the UIC 2. Motion constrained around the cylindrical axis
Pull	1. Two fingertips in contact with the cylindrical surface of the UIC 2. Fingertips move in upward (away) direction with respect to the planar surface of the UIC
Insert (Object)	1. Fingertips in contact with the Object 2. Object is in close proximity with the UIC
Move	1. Two fingertips in contact with the UIC

Table 6.2: Supported simulations and corresponding physical parameters modifications

Supported Simulations		**Physical Parameters Modifications**
Move	General movement	Change 3D position of object
	Along one direction	Constrain the change along a direction vector
	On a 2D plane	Constrain the change on a plane
	Constant speed	Change the 3D position at a constant rate
	With acceleration	Change the 3D position at changing rates
	Move to point	Change the 3D position until it reaches the target
Rotate	General rotation	Modify the rotation matrix
	Single axis	Rotate the object about an axis
	Multiple axis	Rotate the object about each axis and multiply the rotation matrix to obtain the final rotation
	Constant angular speed	Rotate the object at a constant rate
	With acceleration	Rotate the object at changing rates
Size changes		Modify the dimensions of the object
Quantity changes		Add or remove the object from the AR environment
Temperature changes		Modify the RGB colors of the object. Increasing temperature will lead to higher red value and lower green and blue values.

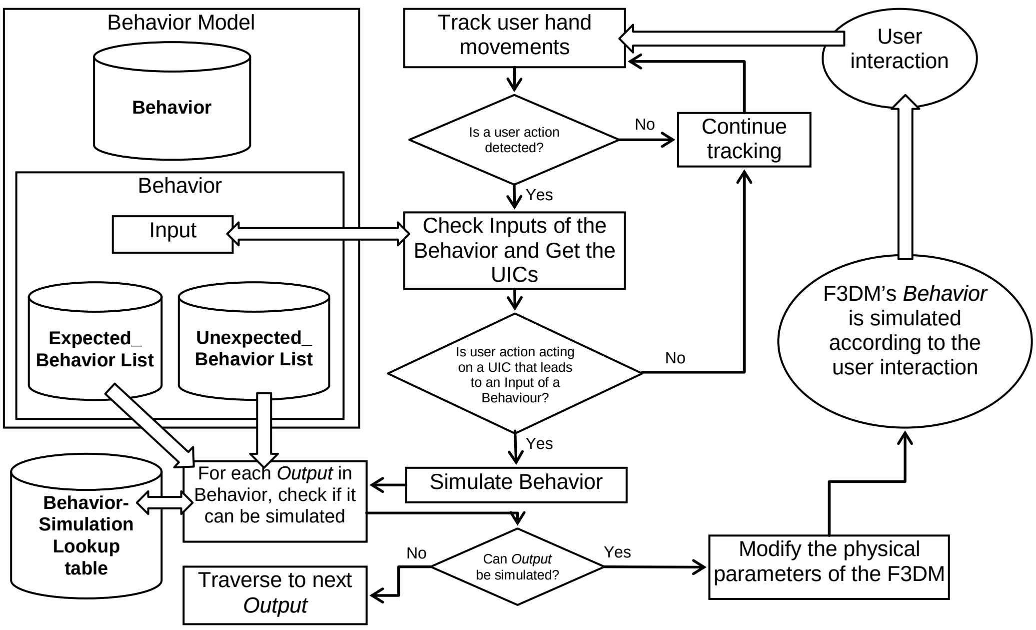


Figure 6.1: Behavioral simulation process

6.3 Functional and Geometrical Design Verification

When a user creates the 3D model of a product, it is possible that the 3D model may be inconsistent with the functions. Therefore, the Archetype Product Model (APM), which inherits the functions, product structure and behavior of the *Product*, is used to generate design rules that must be fulfilled by the design candidates. The design rules will be used to verify whether the product F3DM has the required components to fulfill the designed functions and whether the functions designed for them can be achieved geometrically.

The design rules can be categorized into the functional design rules, which determine the functions that the product must have, and the geometrical design rules that determine the geometrical relationships of the components. The functional design rules are derived from the reasoned functions (*Function_Chains* and *FBS_Primitives*) and used to check whether there is any missing component in the design candidates that is critical functionally. The geometrical design rules are derived from the *Geometrical_Relationships* of the *Object_Pairs* and are used to check whether the components in the design candidates are of the correct shapes, positions and orientations so that they can perform their functions.

From the APM, a design candidate is created as an instance. The design parameters of the components are extracted and evaluated with relevant rules. The design verification process is conducted by inferring whether the design candidate obeys the design rules. If there is a violation, an instance of the violation class will be created and the design candidate will be linked to the violation instance. There are two types of violation classes, namely, **Missing_Components** and **Geometrical_Corrections**. **Missing_Components** is used to store components that are missing in the design candidate. When there is a missing component, the user will be prompted to create and add it to the design candidate. **Geometrical_Corrections** is used to store the geometrical corrections that are required for the design candidate. There are four types of corrections that can be supported:

1. Shape correction, where the user will be prompted to change the shape of a component to the desired shape so that it can fulfill its function.
2. Size correction, where the user has to change the size of the component.
3. Position correction, where the end user has to reposition the component.
4. Orientation correction, where the end user has to reorient the component.

Corrections can be performed by the system automatically or by the user manually. After checking and correcting the design candidate, this candidate will be evaluated again using the APM to ensure that the function model and 3D model are consistent, i.e., there is no *Violation* for the product F3DM.

6.4 Hand Strain and Ease of Handling Design Analysis

The ergonomics of handling a product F3DM can be evaluated by using the BHIM to track and detect hand strains when the user is handling the product F3DM. The hand strains are recorded for each design candidate and the handling ergonomics of the design candidates can be compared based on a Hand Strain Index, which is calculated from the hand strains detected.

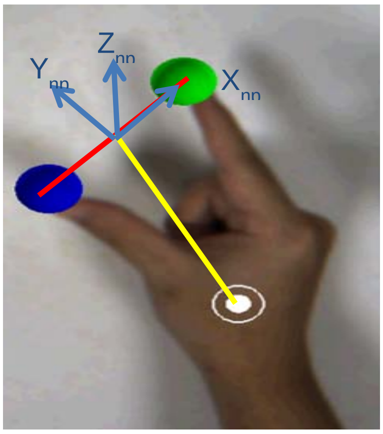
6.4.1 Determination of Hand Strains

Hand strain is defined as the discomfort a user experiences at certain hand postures. Two types of hand strain can be captured. The first type is when the pinch width exceeds 110mm, whereby the user can exert only 60% of the pinch strength (Imrhan & Rahman, 1995). The width of the pinch is defined as the distance between the thumb and the index fingertip. The second type is when the deviation of the wrist angle θ has reached a discomfort range (Khan et al., 2010) as shown in Table 6.3. Figure 6.2 shows the various hand strain postures that can be recorded. Hand strain is detected only when the user is manipulating virtual models. A hand strain is recorded when the hand experiences discomfort for more than 1 second so as to differentiate a strain from a reflex movement, and it contains information on the maximum deviation, the dwell time and the hand in strain.

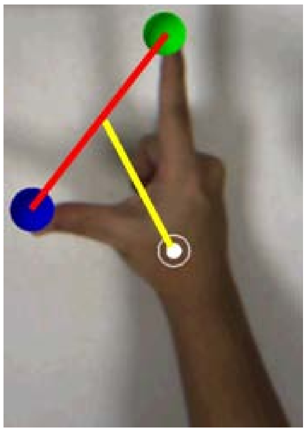
A hand strain is terminated when the deviations are below the defined thresholds. Different hand strains can be detected independently. A posture can be detected to experience three hand strain incidents concurrently, e.g., a wide pinch strain, flexural strain and pronation strain. Studies have demonstrated the effects of combined strains (Khan et al., 2010). However, it is difficult to obtain a formula to calculate the total strain. Therefore, hand strains are treated as independent of each other.

Table 6.3. Discomfort range for different wrist angles (Khan et al., 2010)

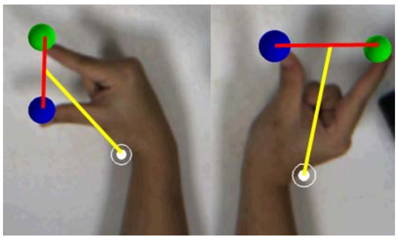
Deviation types	Range of motion (ROM)	Discomfort range	
Flexural	95°	>45% of ROM	>43°
Extension	85°	>45% of ROM	>38°
Radial	45°	>45% of ROM	>20°
Ulnar	70°	>45% of ROM	>32°
Pronation	130°	>45% of ROM	>59°
Supination	145°	>45% of ROM	>65°



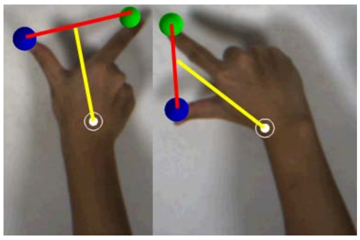
(a) Neutral posture with reference coordinate system



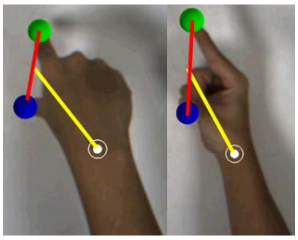
(b) Wide pinch strain



(c) Flexural and extension strains



(d) Ulnar deviation and radial deviation strains



(e) Pronation and supination strains

Figure 6.2: Hand strain postures detected and recorded using ARCADE

6.4.2 Strain from Deviation of Wrist Angle

The wrist angle deviation θ is determined as the rotation from the coordinate systems of the neutral posture CS_{np} of the hand (Figure 6.2a), which is the posture where the bones of the fingers and forearm are roughly parallel (Khan et al., 2010), to a new posture CS_{new}. The flexural/extension (F/E) angle, radial/ulnar deviation (R/U) angle, and pronation/supination (P/S) angle are calculated from the rotations about $\hat{Z}_{new}$ of CS_{new} from $\hat{X}_{np}$ to $\hat{X}_{new}$, about $\hat{X}_{new}$ of CS_{new} from $\hat{Z}_{np}$ to $\hat{Z}_{new}$ and about $\hat{Y}_{new}$ of CS_{new} from $\hat{X}_{np}$ to $\hat{X}_{new}$, respectively. The rotation from CS_{np} to CS_{new}, $R_{CS_{np}\to CS_{new}}$, is a combination of the rotation from $\hat{N}_{np}$ to $\hat{N}_{new}$, $R_{N_{np}\to N_{new}}$, and the rotation $R_{N-\theta}$ about $\hat{N}_{new}$ with θ, as indicated in Equation (6.1), where N represents the corresponding axis to find θ. $R_{CS_{np}\to CS_{new}}$ is derived using Equation (3.2) and θ is derived from $R_{N-\theta}$.

$$R_{N-\theta} = R_{CS_{np}\to CS_{new}} \times R_{N_{np}\to N_{new}}{}^{-1} \qquad (6.1)$$

The P/S angle is θ as pronation and supination only occur at the wrist. For F/E and R/U angles, θ consists of the rotations of the forearm about the elbow joint, which must be eliminated. For F/E angle, ϕ between the two vectors $\hat{V}_{th\to if}$ and $\hat{V}_{hc\to mp}$ is constant when there is only forearm rotation. The new $\hat{X}'_{np}$ without the forearm rotation can be obtained using three simultaneous equations as shown in Equation (6.2) based on three constraints that $\hat{X}'_{np}$ must satisfy, namely, the angle between $\hat{X}'_{np}$ and $\hat{V}_{new_{hc\to mp}}$ must be equal to that between $\hat{X}'_{np}$ and $\hat{V}_{np_{hc\to mp}}$, $\hat{X}'_{np}$

must lie in the $\hat{X}_{new}\,\hat{Y}_{new}$ plane, and $\hat{Z}'_{np}$ must be parallel to $\hat{Z}_{new}$. The F/E angle will be the rotation angle from $\hat{X}'_{np}$ to $\hat{X}_{new}$ about $\hat{Z}_{new}$.

$$\hat{X}'_{np} \cdot \hat{V}_{new_{hc \to mp}} = \hat{X}_{np} \cdot \hat{V}_{np_{hc \to mp}} \tag{6.2-i}$$

$$\hat{X}'_{np} \cdot \hat{Z}_{new} = 0 \tag{6.2-ii}$$

$$(\hat{X}'_{np} \times \hat{V}_{new_{hc \to mp}}) \cdot \hat{Z}_{new} = 1 \tag{6.2-iii}$$

For R/U angles, the ulnar deviation of the forearm is insignificant assuming that the elbow of the user is placed on the table top of the assembly workspace. The radial deviation angle of the forearm, φ can be calculated from the arcsine of the average human forearm length of 26.5cm (Chaffin et al., 2006), over the z-coordinate of $\hat{V}_{new_{hc \to mp}}$. It is subtracted from θ to obtain the radial deviation wrist angle. It is possible that the position of the elbow will change with a rotation of the shoulder. However, rotation of the shoulder cannot be captured by the system and thus there will be an error using the current method, which can be resolved by using more tracking devices.

6.4.3 Calculation of Hand Strain Index

A Hand Strain Index (HSI) is derived from the Strain Index (Moore & Vos, 2004), and it uses three variables, namely, Hand/Wrist Posture, Duration of Exertion and Efforts/Minute to evaluate the hand strains of an assembly step. An assembly step is defined as a single assembly of a component to another component. Table 6.4 shows the rating and multiplier table of the variables used to derive the HSI. The posture rating ranges from fair to very poor as only undesirable hand postures are considered. For each hand strain incident, the percentage strain $\%S_i$ = $\frac{\max dev - threshold}{threshold} \times 100\%$ is calculated. The mean $\%S_i$ for different hand strain incidents in an assembly step is calculated to obtain the posture ratings and multiplier values from the Hand/Wrist Posture column in Table 6.4.

Table 6.4. Rating and Multiplier values for Hand Strain Index used in ARCADE

Rating	Hand/Wrist Posture	$\%S_i$	Duration of Exertion (%)	Efforts/Min
1	Very good (1.0)	-	<10 (0.5)	<4 (0.5)
2	Good (1.0)	-	10-29 (1.0)	4-8 (1.0)
3	Fair (1.5)	0 - 9	30-49 (1.5)	9-14 (1.5)
4	Poor (2.0)	10-19	50-79 (2.0)	15-19 (2.0)
5	Very Poor (3.0)	≥20	≥80 (3.0)	≥20 (3.0)

Note: Multiplier values in parentheses

The Duration of Exertion is the percentage of the total durations of all the hand strain incidents over the total duration of the assembly step, $\frac{\sum duration\ of\ strain\ events}{Total\ time\ taken\ for\ assembly} \times 100\%$. The Efforts/Min is the number of hand strain incidents detected per minute. The HSI is the product of the multiplier values (MV) of the variables, $HSI = MV_{HandPosture} \times MV_{DurationofExertion} \times MV_{Efforts/\min}$. For example, during an operation of assembling a pin into a hole, the user is detected to have undergone two strain incidents over the entire duration of the operation with $\%S_1$ at 6 and $\%S_2$ at 12, which leads to a mean %S of 9 and $MV_{HandPosture}$ of 1.5. The entire duration of the operation is 100 seconds and the duration of S_1 is 18 seconds and S_2 is 13 seconds. This leads to a 31% Duration of Exertion and a corresponding $MV_{DurationofExertion}$ of 1.5. Only two strain incidents have been detected over a period of 100s, and this results in a pro-rated value of 1.2 Efforts/Min and the corresponding value of $MV_{Efforts/Min}$ of 0.5. The HSI is calculated to be 1.5x1.5x0.5=1.125. A Strain Index of 5.0 is considered to be associated with hazardous work (Moore & Vos, 2004). In ARCADE, the aim will be to detect and reduce HSI that exceeds 5.0 during manual assembly.

6.4.4 Detection of Hand Strain Incident during Handling

When the user is handling the product F3DM, a hand strain incident is recorded when a hand strain is detected. Some occurrences of hand strains are shown in Figure 6.3. The types of hand strain, maximum deviation, hand that is in strain, strain duration and the component(s) involved. The HSI is calculated and can be used to assess the ergonomics of different design candidates.

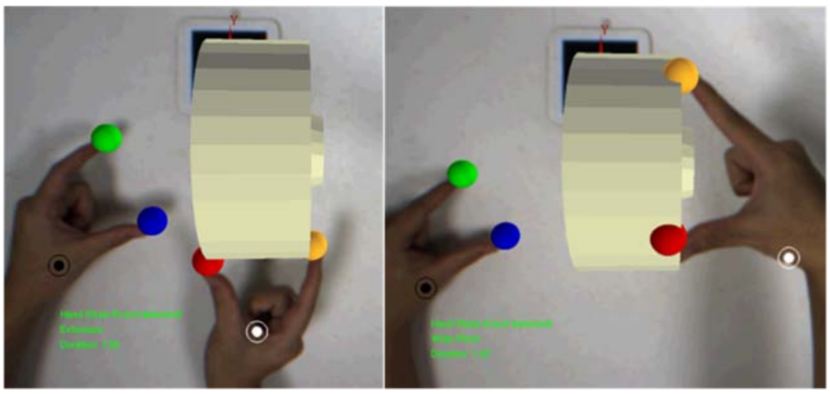


Figure 6.3: Example of hand strain when handling a product F3DM

The hand strain detection in ARCADE considers assembly ergonomics, which is not captured using conventional design evaluation tools. The HSI provides an overview of the strain sustained when the user is handling the product and the individual hand strain incidents can be analyzed to understand the causes and determine the solutions to eliminate them, e.g., improving the component access or modifying the size of the component.

7. Case Studies

7.1 Introduction

Three conceptual design case studies have been conducted using ARCADE to demonstrate its application for conceptual design.

The first case study involves the design of a desk-top cleaner which will clean the table via user command. A new user interaction has been designed which consists of using the finger to point at the region where the desk-top cleaner has to clean. The F3DM of the desk-top cleaner is created and the actual use of the desk-top cleaner is simulated in ARCADE.

The second case study involves the design of a fruit processor which is able to peel, cut and blend the fruit into fruit juice. In particular, conceptual design is performed for a fruit-peeling and cutting module. The functional and geometrical aspects of the fruit-peeling and cutting module are explored in ARCADE to develop feasible peeling and cutting processes. They are then simulated for the user to evaluate and improve the design.

The third case study involves the design of an electric toy car that can be assembled by the user. The 3D models are created using ARCADE and detailed design is performed using conventional CAD software. The design is evaluated on the ease of handling and assembly.

7.2 Case Study 1: Design of a Table-top Cleaner

A case study on the conceptual design of a table-top cleaner (TC) has been conducted. The conceptual design of a TC can be considered as a design of a new-to-the-world product. The main function of the TC is to clean the top of a table, which means the removal of dirt from the table-top, on the command of the user. There is a need to understand how the user is going to use a TC. A F3DM model of the TC is created using ARCADE so that the designer can work with a functional prototype and present the concept of a TC to the user. The conceptual design process begins with the designer defining the PUM, creating the 3D model of the TC in ARCADE and testing the F3DM as a functional prototype. The function, structure and behavior reasoning processes are performed at the back-end to create the F3DM for testing.

7.2.1 Defining the PUM and Reasoning the Functions

To fulfill the function of cleaning the table-top, the designer has defined three functions for the PUM, namely:

1. Press – Button → TC– Clean Table-top
2. Fingertip – Fingertip sensor → TC – Move To Fingertip
3. Edge – Edge sensor → TC – Stop

The first PUM function means that the TC can be switched on by pressing a button, with “Press-Button” as an *Input*. The *Output* of “TC-Clean Table-top” is classified as a task *Output* that can be decomposed into “TC-Move” and “TC-Remove Dirt”.

The second PUM function means that the TC will sense the fingertip of the user using a fingertip sensor (FS), resulting in an *Input* “Fingertip-FS”, and move to a location specified cleaning the path that it has travelled simultaneously. This allows the user to instruct the TC to clean certain areas of the table-top, which is rather new and there is a need to understand how the user will use the TC. The *Output* of “TC-Move to Fingertip” is a task *Output* that can be decomposed into “TC-Move” and “TC-Fingertip position”.

The third PUM function is a passive function where the TC will sense an edge using an edge sensor (ES), resulting in an *Input* “Edge-ES”, and stop moving to prevent it from falling off the table-top. “TC-Stop” is the *Output* that will stop the TC when it reaches the edge of the table-top.

With these initial three functions, there are a total of three initial *Inputs*, “Press-Button”, “Fingertip-FS” and “Edge-ES” and four initial *Outputs* “TC-Move”, “TC-Remove Dirt”, “TC-Fingertip position” and “TC-Stop” to be reasoned. Four *Function_Chains* are created from the initial *Inputs* and *Outputs*. The first *Function_Chain* has an *Input* “Press-Button” and *Output* of “TC-Move”. The second *Function_Chain* has an *Input* of “Fingertip-FS” and *Output* of “TC-Target position” since the other sub *Output* of “TC-Move” can be fulfilled by the first *Function_Chain*. The third *Function_Chain* has an *Input* of “Edge-ES” and *Output* of “TC-Stop” as defined by its PUM. The fourth *Function_Chain* has an *Output* of “TC-Remove Dirt” and an unknown *Input*. The *Function_Chains* are reasoned sequentially in ARCADE to establish their *Bodies*. However, the fourth

Function_Chain has no prior solution and a *FBS_Primitive* "Brush-TC-Remove Dirt", which is derived from using a brush to remove dirt, has been created. Figure 7.1 shows the functional reasoning processes from the PUM to the final results of the reasoned functions of the TC.

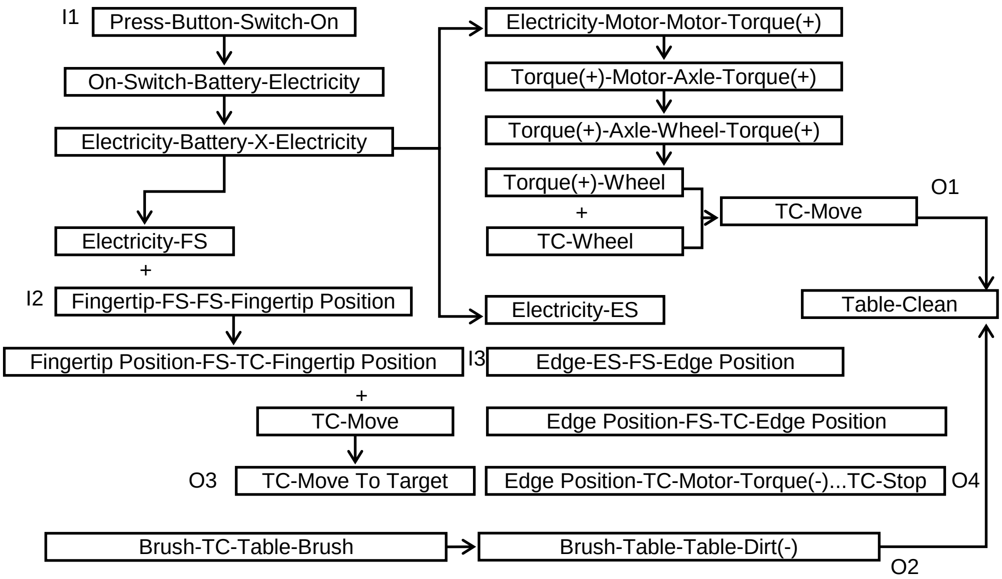


Figure 7.1: Reasoned functions of the table-top cleaner

From Figure 7.1, it can be seen that "Press-Button" is the *Input* to switch on the TC where electricity is provided to the sensors. It is linked to the "TC-Move" *Output* using the causal reasoning process of the FBSML framework, which establishes that the TC is moved by the wheels attached to it, which are driven by a motor.

"Fingertip-FS" requires the "Electricity-FS" sub-*Input* to achieve the "TC-Fingertip position" *Output* to determine the fingertip position. This is combined with “TC-Move” to obtain the “TC-Move to Fingertip” task *Output* as defined by the second PUM function.

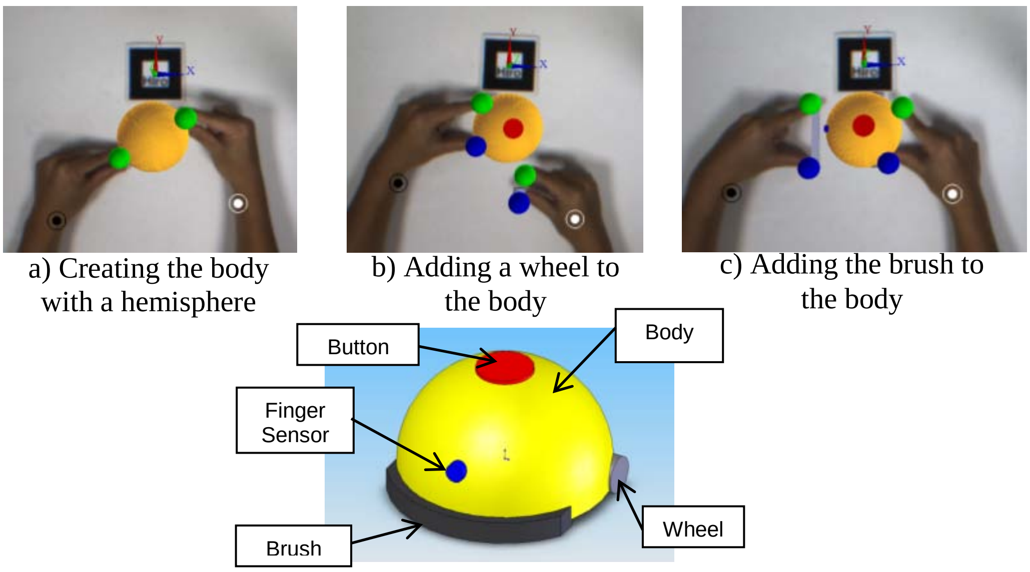
“Edge-ES” is combined with “Electricity-ES” sub-*Input* to sense the position of the edge, which leads to the “TC-Stop” *Output*, which is preceded by the removal of torque to the wheels and motor. This is a reverse of the *FBS_Primitives* that provide torque to the wheels to move the TC for the “TC-Move” *Output*.

7.2.2 Generating the 3D model and Design Verification

From function reasoning, the required components of the TC can be derived and they are a main body, button, wheels, brush, finger sensor, edge sensor, battery, motor and axles. A few components are visible to the user while others are hidden in the main body. Therefore, not all the components need to be represented as 3D models at the initial design stage.

The designer creates 3D models for the main body (hemisphere), button (disc), wheels (cylinder), brush (block) and finger sensor (cylinder). The basic shapes of the component are in parentheses. The user begins by defining the size of the hemisphere for the main body and adding the other components onto it through directly manipulating the 3D models. Figure 7.2 shows a few screenshots of the design process and the final 3D model of the TC, with the components aligned correctly. As the brush does not fit the round shape of the main body, it is modified in SolidWorks to an arc shape. The 3D models created in ARCADE can be used seamlessly for detailed design using conventional CAD software.

The structure reasoning process for this particular design of the table-top cleaner is as follows. From the function model in Figure 7.1, the only mechanical type of *Contact* that occurs for all the *Object_Pairs* in the design is found in the "Motor-Axle" and "Axle-Wheel" *Object_Pairs*. This is due to "Torque-Torque" *Functional_Relationship* that the *Objects* shared, which lead to a "Mechanical_Contact_Coaxial" *Contact*. Since they have a "Mechanical_Contact _Coaxial" *Contact*, they are inferred to have a "Geometrical _Relationship_Coaxial" and "Geometrical_Relationship_Rigid_Joint" *Geometrical_Relationships*. This dictates that the 3D model of the wheel ,axle, and motor must be coaxial and the wheel must be connected rigidly to the axle and the motor. Since there is no 3D model of the motor and axles and there are two wheels for this particular design of the table-top cleaner, this leads to a single geometrical design rule that defines that both wheels to be coaxial.



a) Creating the body with a hemisphere

b) Adding a wheel to the body

c) Adding the brush to the body

d) Final 3D model of the table-top cleaner

Figure 7.2: Generation of the table-top cleaner's 3D model

7.2.3 Testing the F3DM as a Functional Prototype

After generating the function model and the 3D model of the TC, the F3DM of the TC is generated and the designer can evaluate the product through simulating it in the AR design environment. The TC will behave according to the PUM during simulation. Three types of behavioral simulations are derived from the behavior model of the F3DM:

1. The TC will be switched on with the pressing of a button and remove dust from the table-top in areas that it has travelled (*Input*: “Press-Button”, *Expected_Behavior*: “TC-Move”, “TC-Remove Dirt”).
2. The TC will move to the fingertip when the fingertip is in view (*Input*: “Fingertip-FS”, *Expected_Behavior*: “TC-Move To Fingertip”).
3. The TC will stop moving when it reaches the edge of the table-top (*Input*: “Edge-ES”, *Expected_Behavior*: “TC-Stop”).

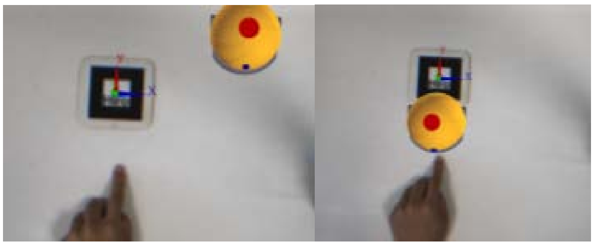
For the first behavioral simulation, ARCADE will detect and track the fingertip to check if it has touched the 3D model of the button. When the fingertip is detected to have touched the button and moved in a downward direction, the *Input* of “Press-Button” is determined to have occurred. This will trigger the TC to start moving and to remove dirt in the area that it has passed. Simulation of the movement of the TC is achieved through modifying its 3D position in the direction that it is travelling. In order to simulate the removal of dust on the table-top, virtual dust particles are placed on the table-top. When the TC passes areas with dust particles, the dust particles will disappear to simulate the cleaning of the table-top by the TC as shown in Figure 7.3.

For the second behavioral simulation, ARCADE will track the fingertip and determine its 3D position relative to the current location of the 3D model of the TC. The fingertip position will be the target position that the TC must move to. The TC will rotate to face the fingertip and move towards it. During this behavioral simulation, the user can point at a certain location and the TC will move towards the fingertip as a form of control over the TC.

For the third behavioral simulation, the 3D position of the TC is tracked by ARCADE and compared with the known boundary of the table. If the TC is tracked to have reached the edge of the table, there will not be any more changes to its 3D position. During this behavioral simulation, the TC is constrained to move within the table-top and will stop when it reaches the edges of the table-top.

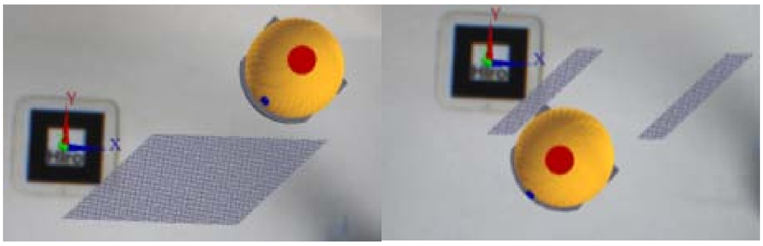
In the absence of actual fingertip and edge sensors, the behavior of the TC can still be simulated in ARCADE. This allows the designer to be able to work with a functional prototype. Together with the simulation of using the fingertip to control the TC, the entire cleaning process using the TC is simulated. Figure 7.3 shows the simulations of the functions of the TC and their corresponding Behavior derived from the F3DM.

By creating a F3DM of a TC, the designer can test a functional prototype and present the design concept to the other stakeholders such as the final user of the product. With a F3DM, the designer can play with the design and make improvements to it. For example, when testing the TC F3DM, the designer observes that the TC does not have any solution for navigating around obstacles on the table-top and this can be added as a function in the next design iteration.



a) Moving to the fingertip

b) Stopping at the edge



c) Cleaning the desk

Figure 7.3: Simulation of the behavior of the table-top cleaner

This case study demonstrated the application of F3DM in the design of a new-to-the-world product. It is unclear to the designer and user how such products can be utilized in the use environment. By creating the product F3DM, the actual usage of the TC in a common desktop environment can be simulated and the designer and user can evaluate the design by interacting with it. This allows the designer to demonstrate the TC concept. In addition, the designer can explore and understand the TC design better and improve it, i.e., adding functions to navigate past obstacles.

7.3 Case Study 2: Design of a Fruit Processor

This case study involves the conceptual design of a fruit processor (FP) that is able to peel, cut and finally blend the fruit into juice. There are existing juicers which blend sliced fruits into juice and the main objective of the FP is to automate the fruit peeling and cutting processes. The designer has to design a fruit peeling and cutting (FPC) module and adds it to an existing juicer design. There is a need to understand the functional and geometrical aspects of the FPC module in order to come up with a good conceptual design and this case study demonstrates the role of F3DM in helping the designer achieve this need.

7.3.1 Defining the PUM and Reasoning of the Functions

The designer begins with defining the PUM of the FP. This is rather straightforward, as the user just has to insert a fruit into the fruit container of the FP and select the mode to process the fruit. The PUM has the following functions:

1. Insert-Fruit + Press-'Peel' Button → Fruit Container – Fruit (Peeled)
2. Insert-Fruit + Press-'Cut' Button → Fruit Container – Fruit (Cut)
3. Insert-Fruit + Press-'Juice' Button → Juice Container – Fruit Juice
4. Press-'On' Button → FP – On

The first three PUM functions require two *Inputs*, "Insert-Fruit" and "Press-'X'Button" where 'X' indicate the mode that the FP should be working in. There are three modes. The 'Peel' mode will peel the fruit in the fruit container of the fruit processor and lead to the "Fruit Container-Fruit (Peeled)" *Output*. The 'Cut' mode will cut the peeled fruit in the fruit container into pieces and result in the

“Fruit Container-Fruit (Cut)” *Output*. The ‘Juice’ mode will send the peeled and cut fruit pieces in the fruit container to the juicer module of the FP and blend them to obtain fruit juice in the juice container. This leads to the “Juice Container-Fruit Juice” *Output*. The fourth PUM function is to represent the turning on of the FP with an Input of “Press-‘On’Button” and Output of “FP-On”

The peeling, cutting and juicing process are sequential, i.e., peeling has to be performed before cutting and peeling and cutting must be performed first before blending. As the designer does not want to redesign the juicing process, the juicer is represented as *FBS_Primitives*: “Fruit (Cut)-Juicer-Juice Container-Fruit Juice” and “Fruit (Cut)-Juicer-Waste Container-Pulp”. The former represents the process of extracting juice from the fruit using the juicer and the latter represent the side effect of producing pulp when juicing. Therefore, there is no need for function reasoning of the juicing process and this allows the designer to concentrate on the function modeling of the FPC module.

As there is no automatic peeler or cutter in the product database, the designer has to study the function models of manual peeling and cutting. The manual peeling process can be modeled as either “Blade (Move)-Fruit-Fruit-Fruit (Peeled)” or “Fruit (Move)-Blade-Fruit-Fruit (Peeled)” with the side effect of “Fruit-Fruit Skin” as an *Output* and the user providing the mechanical energy to move the blade and fruit. Similarly for the manual cutting process, the function model is similar with the difference in the output being “Fruit-Fruit (Cut)”. Conceptually, the peeling and cutting processes can be abstracted into providing certain movements to either a blade or a fruit and putting them in contact to achieve their respective final *Outputs*.

For function reasoning, the goal is to determine a functional design that is able to achieve the *Outputs* of “Fruit-Fruit (Peeled)” and “Fruit-Fruit (Cut)”. These *Outputs* can only be derived from the user-defined *Inputs* of “Blade (Move)-Fruit” and “Fruit (Move)-Blade” as there is no prior *FBS_Primitives* that can achieve the *Outputs*. Therefore the reasoning process is used to find a solution that can achieve the *Outputs* of “Blade-Blade (Move)” and “Fruit-Fruit (Move)”. This means that the blade and the fruit will have to be moved automatically to come into contact to achieve the *Inputs* “Blade (Move)-Fruit” and “Fruit (Move)-Blade” in order for the fruit to be peeled and cut. The function reasoning process thus begins with an initial *Input* of “Press-Button”, which turns on the FP, and initial *Outputs* of “Blade-Blade (Move)” and “Fruit-Fruit (Move)”.

From the function reasoning process, a possible solution defined by the designer for the peeling process is to use a shaft to rotate the fruit, which is derived from the function model of a rotary motor rotating an object such as a wheel, and a linear motor to move the blade, which is derived from the function model of a linear motor moving an object such as the opening of a CD tray, towards the fruit. The function model for this solution is shown in Figure 7.4. Functionally, it is possible to cut the fruit in the same manner and in the first design iteration, the designer uses the same function model for both the peeling and cutting process.

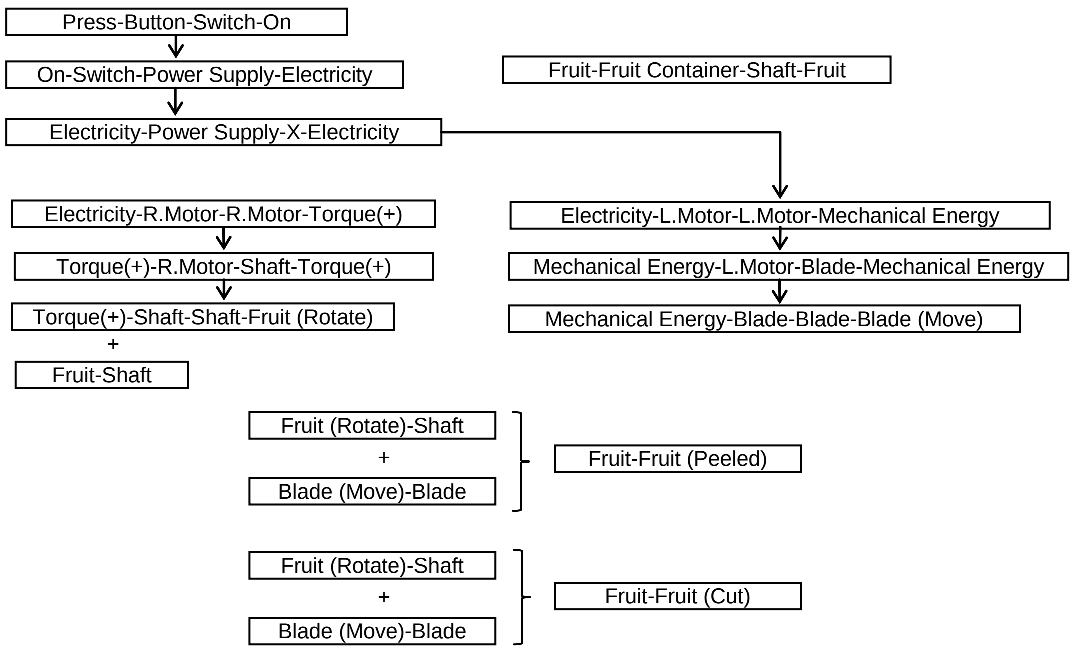


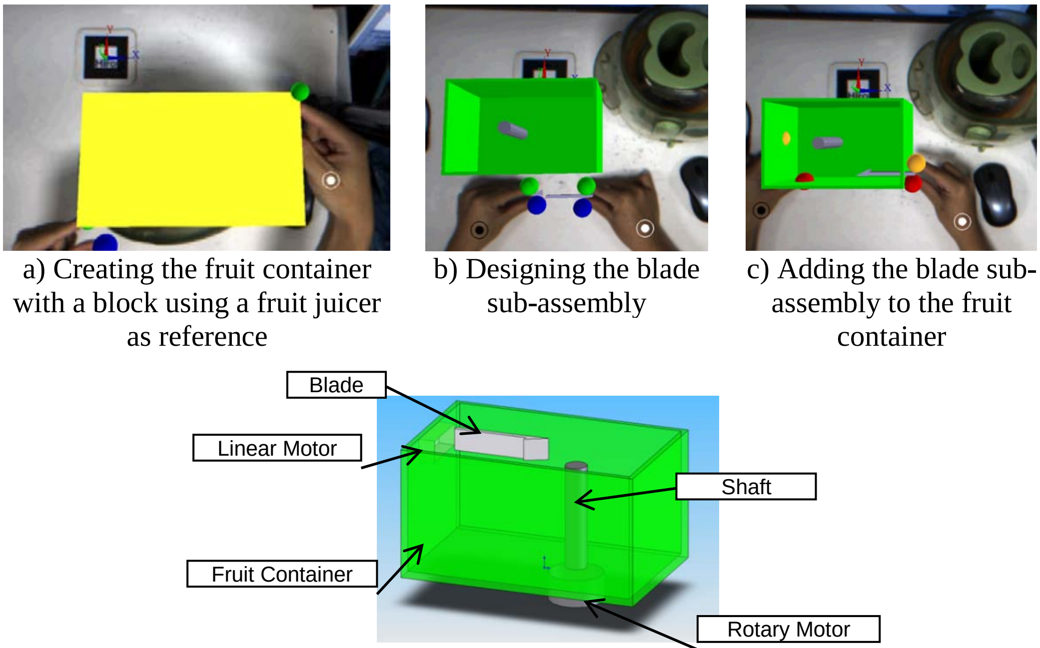
Figure 7.4: Function model of FPC module

7.3.2 3D Design

From function reasoning, the required components for the FPC module include a fruit container, a shaft, a blade, a motor, a linear motor and a power supply. As a physical juicer is available, the designer only has to generate the 3D model of the FPC module in ARCADE. He can use the juicer as a reference for the dimensions of the FPC module. Figure 7.5 shows a few screenshots of the 3D design process.

7.3.3 Behavior Simulation and Design Evaluation

The user interaction with the FP is rather straightforward with the user inserting the fruit and pressing the button to select the desired mode, and thus simulation is not required for such actions. The main focus of behavior simulation for this case study is to understand the peeling and cutting processes in the FPC module. While



a) Creating the fruit container with a block using a fruit juicer as reference

b) Designing the blade sub-assembly

c) Adding the blade sub-assembly to the fruit container

d) Final 3D model of the FPC module

Figure 7.5: 3D design of the FPC module

the system is able to simulate the movements of the fruits and blade, it is not able to simulate the effect of cutting and peeling generically.

There are two behavior simulations are as follows:

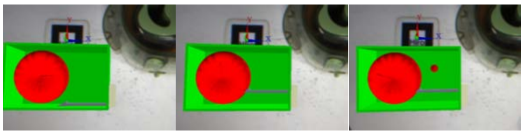
1. The linear movement of the blade driven by the linear motor is simulated. The blade will be simulated to move in a single direction derived from the 3D model of the FPC module. The blade will move towards the fruit and be in contact with it to achieve the "Blade (Move)-Fruit" *Input* required to peel and cut the fruit.

2. The rotation of the fruit by the shaft is simulated. The shaft will rotate the fruit when the blade is in contact with the fruit to achieve the "Fruit (Move)-Blade" *Input*.

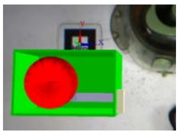
During behavior simulation, ARCADE will track the position of the blade with respect to the fruit and perform the two simulations accordingly. From the function model, there is no constraint on the placement of the shaft and blade and the fruit will be peeled and cut as long as the blade and fruit are in contact and either the blade is moving or the fruit is rotating. However, this is not true from the geometrical aspect of the design and the manner in which the blade and fruit move with respect to one another will determine whether the fruit is cut or peeled. The behavior simulation process in ARCADE allows the designer to visualize the designed peeling and cutting process and this functional-geometrical design issue, where the function model is not able to differentiate the geometrical aspect of the design can be spotted by the designer. Therefore, the designer will have to design the geometrical aspect of the peeling and cutting processes and simulate them to arrive at a feasible solution for the FPC module.

From the behavior simulation of the peeling process, the designer realizes that the blade is lacking one axis of motion in order to peel the fruit completely (Figure 7.6a). This means that there is a need to add another linear motor to move the blade along another axis. In addition, there are unwanted fruit skin left from the peeling process and this means that the designer must design a process for removing them before proceeding to the next step of cutting the fruit.

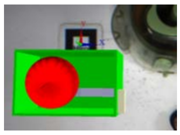
In the design of the cutting process, the designer can manipulate the blade and fruit directly and play around with them to come up with a new cutting method (Figure 7.6b). In the new design, the fruit will be cut horizontally first with the blade piercing the fruit at a fixed location and the shaft rotating the fruit. This is followed by vertical cut where the shaft will rotate the fruit to a certain position and the blade will cut the fruit in a top-down direction (Figure 7.6c).



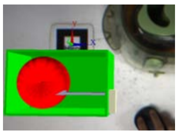
a) Simulation of the peeling process. Only one segment of the fruit is peeled as the blade cannot move vertically



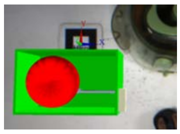
(i) Rotate the blade to the horizontal position



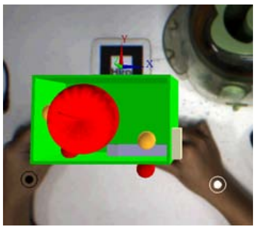
(ii) Pierce the fruit and rotate the fruit for horizontal cut



(iii) Rotate the blade to the vertical position and above the fruit



(iv) Move the blade down for vertical cut



b) Direct manipulation of the blade to explore the cutting process

c) Simulation of the new cutting process

Figure 7.6: Behavior simulation of the FPC module

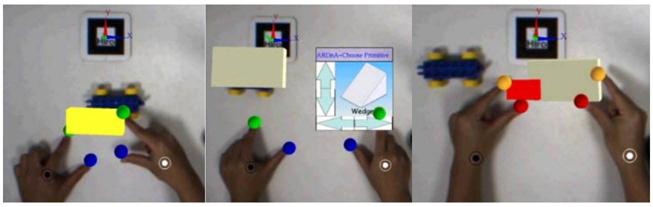
From the behavior simulation of the peeling and cutting processes, the designer is able to identify inconsistencies between the function model and the geometrical model and utilize these findings to improve the design of the FPC module. More components (linear motors) are added as a result and the effect of the placement of the shaft and blade has been emphasized by using F3DM for conceptual design. This case study demonstrates the role of F3DM and the DSVEM in helping the designer understand the functional and geometrical aspects of a design. The peeling and cutting process for the FPC module is complicated and it is possible for the design concept to be functionally feasible but geometrically infeasible as evident in the initial function model. The F3DM allows the designer to play around with the 3D models of the FPC and observe the results via the behavioral simulation. This leads to a better design of the FPC module, which is functionally and geometrically consistent.

7.4 Case Study 3: Design of an Electric Toy Car

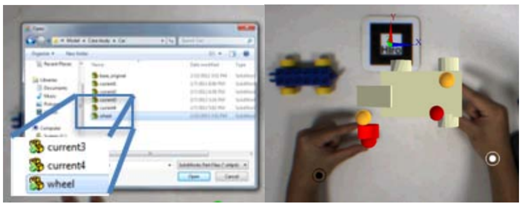
A case study on the design and assembly of an electric toy car was conducted using ARCADE. The main purpose of this case study is to demonstrate the application of ARCADE for evaluating the handling process and ergonomics of a design. Therefore, the function model of the toy car is not created compared to the first and second case study. The designer will only create the 3D model of the toy car and evaluate the assembly and handling processes in this case study. The basic design requirements are the car must be of similar size to an existing toy car, a predefined electric motor must be used and it must be easy to assemble.

7.4.1 3D Design of the Toy Car

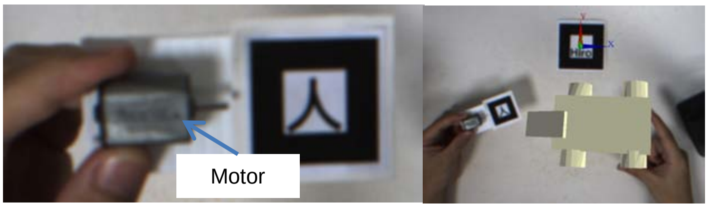
The designer starts by creating a block primitive as the body using an existing toy car as a spatial reference. The front nose is created using a wedge and the wheel using a cylinder. As more than one wheel is required, the wheel is duplicated and placed at different locations. The 3D model of the motor is added using a real motor. The real motor is tracked in the ARWCS by affixing marker onto it (Figure 7.7c). The designer manipulates the real motor and adds a 3D model of it to the virtual model of the car, using both real and virtual objects. Finally, a spoiler (a wedge) is created and placed on top of the motor. An initial assembly sequence of body-nose-wheel1-wheel2-wheel3-wheel4-motor-spoiler is generated. Figure 7.7 shows the design process of the toy car.



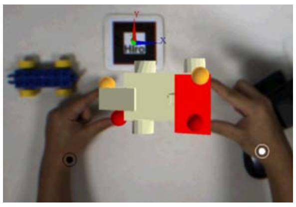
(a) Creating a block for the chassis and a wedge for the nose



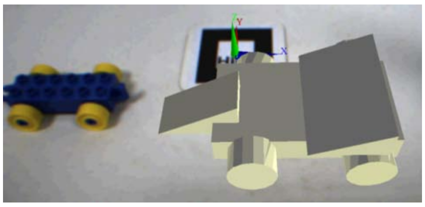
(b) Loading and adding the wheel to the toy car



(c) Adding the real motor to the toy car



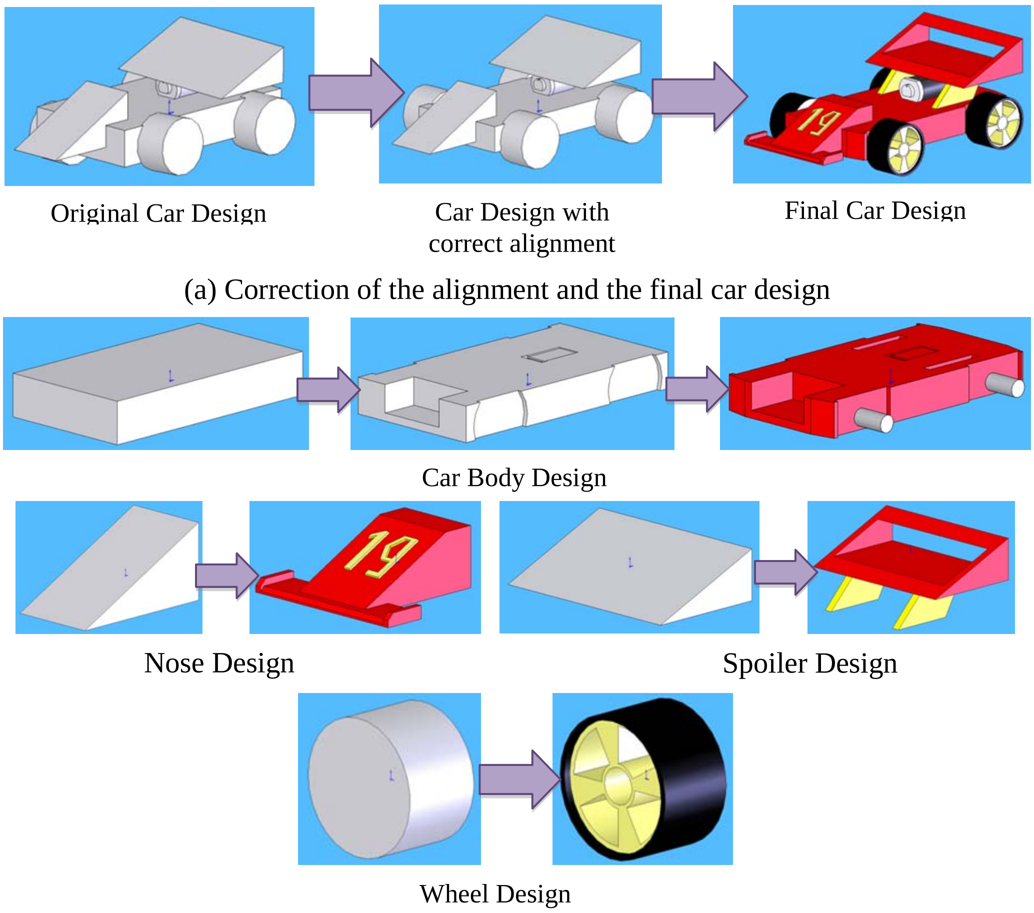
(d) Adding the spoiler to the toy car



(e) The completed toy car

Figure 7.7: Design generation of a toy car in ARCADE

Figure 7.8 shows that the car design has misaligned components. This is due to the inability of the human hands to place the objects precisely. This is corrected by modifying the component placements, followed by the detailed design in CADM (Figure 7.8). For the body, slots are added to insert the spoiler and shafts for connecting the wheels. For the nose, a base is added. For the wheel, a rim is added and a hole is created to fit the shaft. The motor is not redesigned as it is predefined. The spoiler is added with legs so that it can be slotted into the body. The assembly model of the car is generated with the defined mating constraints.

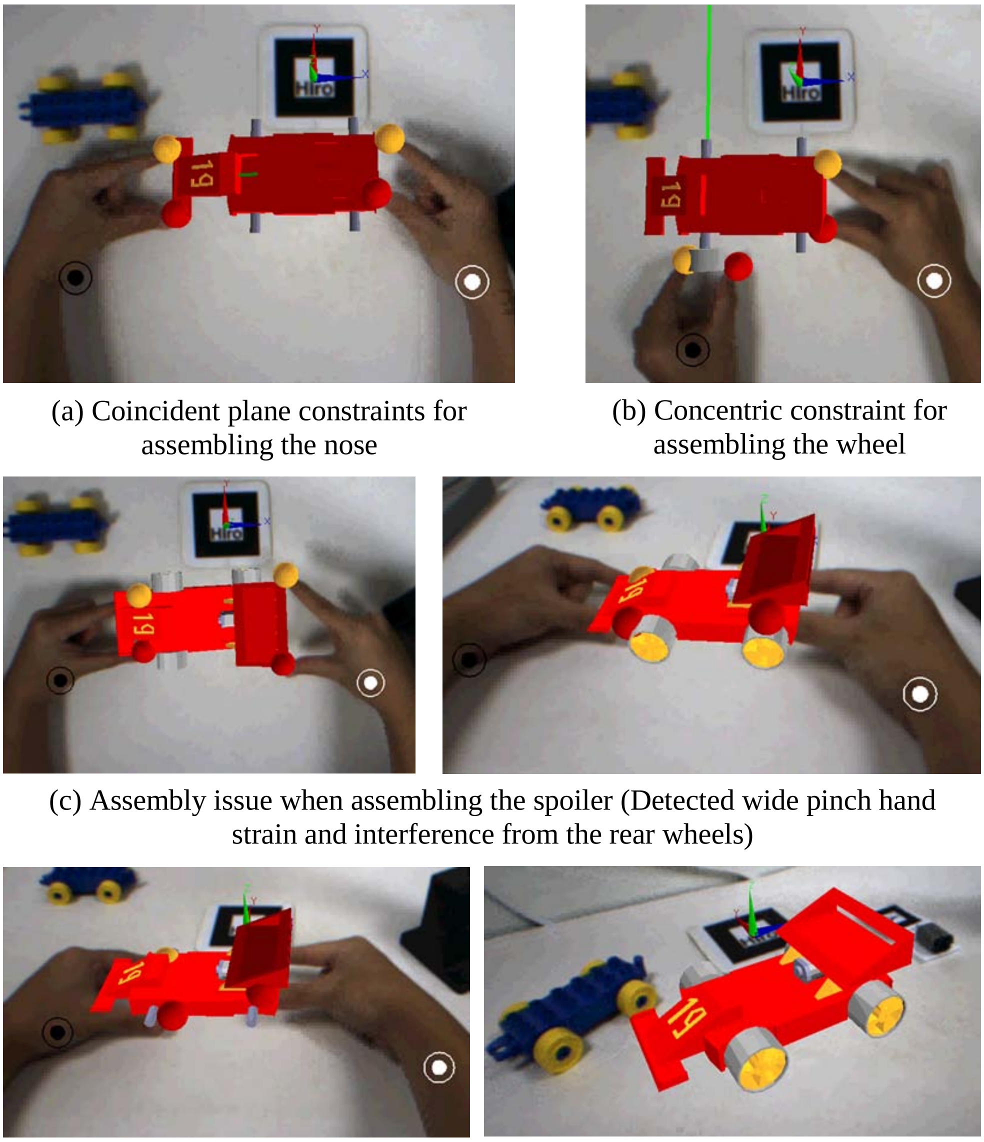


(b) Detailed design of the individual components

Figure 7.8: Detailed design of the toy car

7.4.2 Design Evaluation

The design evaluation is conducted on the assembly and handling of the toy car. The assembly process commences when the body and nose are rendered in ARCADE and the user will start to manipulate the nose and body to fit with one another. . When three surfaces of the nose are constrained with those of the body, the system will check the pose of the nose. If it is the correct pose, the next component can be rendered and assembled. The user will grab the other components and assemble the wheels, motor and spoiler sequentially to form the toy car (Figure 7.9). During the assembly process, the hands of the user are tracked in ARCADE to detect for hand strain incidents using the method described in Section 6.4, so that ergonomics issues with the assembly and handling can be identified. When the assembly is completed, the evaluation results are displayed.



(a) Coincident plane constraints for assembling the nose

(b) Concentric constraint for assembling the wheel

(c) Assembly issue when assembling the spoiler (Detected wide pinch hand strain and interference from the rear wheels)

(d) Alternative sequence of assembling the spoiler before the rear wheels

(e) Assembled toy car

Figure 7.9: Design evaluation of assembly process of the toy car

During the assembly of the spoiler, a hand strain incident is recorded and the strain parameters are: maximum deviation of 125mm, strain duration of 7s and assembly time of 24s. The $\%S_1$ is $\frac{125-110}{110}\times 100\% = 14\%$, which maps to a bad posture and a multiplier value of 2.0. The duration of exertion is $\frac{7}{24}\times 100\% = 30\%$ which has a rating of 3 and multiplier value of 1.5. The effort per minute is $\frac{1}{24}\times 60 = 2.5$ which has a rating of 1 and multiplier value of 0.5. The HSI is calculated to be $2.0\times 1.5\times 0.5 = 1.5$. The strain is caused by the rear wheels blocking the access to the body. Therefore, the sequence has been changed to body-nose-motor-spoiler-wheels. Table 7.1 shows the comparative results of the two assembly sequences.

Table 7.1. Comparative results of two assembly sequence used in case study

	Initial assembly sequence	Amended assembly sequence
Sequence	body-nose-wheel1-wheel2-wheel3-wheel4-motor-spoiler	body-nose-motor-spoiler-wheel1-wheel2-wheel3-wheel4
Total Time taken (s)	90	78
No. of orientation changes	1; At body-wheel3 Body (180°)	1; At body-wheel3 Body (180°)
No. of errors	0	0
HSI	1.5; At body-spoiler Wide pinch strain of 14 %S of duration 7s	0

This case study demonstrated the creation of 3D models using the intuitive 3D modeling module in ARCADE. The created 3D models can be integrated with conventional CAD software to perform detailed design. The final design can be evaluated in ARCADE based on the ease of handling and assembly. Assembly information such as the time taken, orientation changes and errors are captured by the system. In addition, ergonomics information, in the form of the Hand Strain Index, can be detected and analyzed using ARCADE. This is achieved without the production of the actual prototype and allows ergonomics issues to be spotted and addressed early during conceptual design. The user can compare this information between different designs for evaluation.

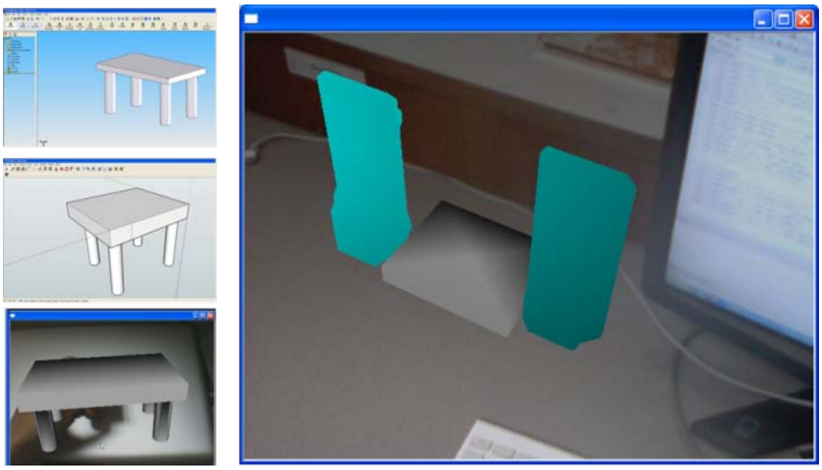
8. User Studies

User studies have been conducted for ARCADE. A preliminary user study has been conducted on the earlier version of ARCADE to compare it with conventional CAD software for creating 3D model. In addition, the bare hand interaction in ARCADE has been tested informally for user feedback on the system. A formal user study has also been conducted on the final ARCADE system to test its intuitive 3D modeling methods, the interactive F3DMs and the design evaluations that are provided.

8.1 Preliminary User Study on Earlier Version of ARCADE

8.1.1 Design Task and Participants Profile

Seven individuals with no CAD modeling experience participated in the user study to compare the earlier version of ARCADE with conventional CAD software. Their ages range from 21 to 35 years. They use computers frequently but are not familiar with 3D modeling applications. In the study, each participant was tasked to create a specified design of a table and a music dock, as shown in Figure 8.1, using ARCADE. A short training session was conducted before the actual modeling and the participants were guided on the different modeling operations during this session.



(a) (b)

Figure 8.1: (a) Table designed using (from top) SolidWorks, Google SketchUp 7 and ARCADE; (b) music dock designed using ARCADE

In the creation of the table, a comparison study was performed with two CAD tools, namely, the Google SketchUp and SolidWorks with prior training sessions for both tools. The time taken by the participants using both tools was recorded for comparison with ARCADE. A qualitative interview was conducted with the participants after the modeling tasks to understand the participants' general impressions, benefits and limitations of ARCADE.

For the music dock, the participants only needed to complete the modeling and there was no recording of the time taken. The main purpose for this part of the user study is to allow the users to reconstruct an everyday object, namely a speaker, and use it to create a music dock, which can be contextualized in a desk top environment. Qualitative reviews of the modeling approach had been obtained from the users.

8.1.2 Results

Figure 8.2 shows the time taken by the participants to model the table. It is evident that less time was needed to create the designs using ARCADE as compared to the other two CAD tools. The use of SolidWorks required the longest (160.6s) and this is mainly due to the use of 2D interfaces to create 3D models. The use of SketchUp required less time from the users as compared to the use of SolidWorks (89.7s) as SketchUp is more intuitive. SketchUp retains the use of using 2D sketches to define the profile of the 3D models and uses dragging and smart selection of the faces. ARCADE requires the least time to complete the table (55.0s). This is mainly due to the use of 3D interfaces to perform some of the modeling operations and the use of AR in visualizing the 3D models during design generation. For example, during the modeling of the table top, only a single operation of positioning two different markers in the 3D design space is required. This is faster than using a 2D sketch followed by an extrusion, which was the case in SketchUp and SolidWorks. In terms of feature creation, ARCADE uses the same methodology as SketchUp and SolidWorks. However, the use of tangible interaction tools in an AR environment allows the user to be able to better visualize and place the features at the desired positions, leading to faster 3D modeling.

In the qualitative interview, some of the responses and general impressions are that ARCADE is intuitive and interactive. Two of the participants commented that ARCADE reminds them of the “World Builder” (Branit, 2007). In terms of benefits, the participants felt that ARCADE is simple to use, e.g., “I only need to move two markers in the design space to create a 3D model", easy to setup, e.g., "the system uses hardware that can be found in any home", interactive, e.g., "I can move the 3D model simply by moving the markers", and fast, e.g., "Drawing in 3D is faster than using a mouse". The use of everyday objects in creating new designs as demonstrated in the music dock user study is also commended by some participants, e.g., "I can use a physical model that is already available for my design".

For the limitations, the participants critiqued that the software is highly dependent of the lighting conditions and shadows, e.g., "Shadows cause the 3D model to disappear during modeling". The jittering of the 3D models, e.g., "The model jumps about at times" and the possibility of strain and fatigue due to prolonged usage, e.g., "Having your hands hanging in the air for a longer time may cause fatigue" were also mentioned by the participants.

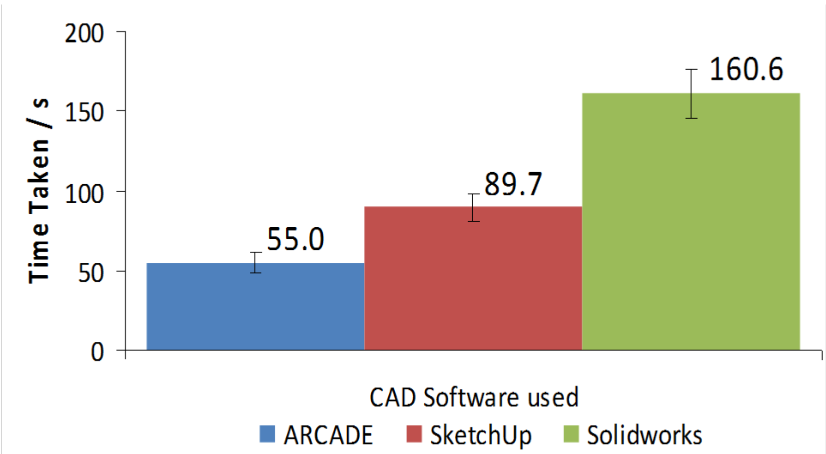


Figure 8.2: Comparison results of SolidWorks, SketchUp and ARCADE in creating a simple table

8.1.3 Discussion

Results from the preliminary user study have indicated that the interactive 3D modeling provided by ARCADE is simple and fast for creating new 3D designs. This implies that a layman will be able to use ARCADE for product design with a relatively low learning curve. In addition, ARCADE is inexpensive and easy to set up. This means that it can be implemented anywhere with only a laptop and a web camera, allowing the user to contextualize the design wherever desired.

The observed ease of using tangible markers to perform 3D modeling operations indicates that interaction tools that provide 3D input information are more suitable for creating new designs. Most modeling operations require input of 3D information from the users and the 3D interaction tools greatly reduce the load on the users in defining the 3D information as compared to 2D interaction tools such as a mouse. In addition, AR gives the users a better understanding of the physical environment and spatial constraints can be taken into consideration during the creation of the design. Therefore, AR interaction tools offer the benefits of being able to provide 3D input and real-time contextualization of the physical environment, which are highly desirable for design tasks.

ARCADE uses everyday objects to create new designs by employing a vision-based system to track, reconstruct and register information on the objects of interest. This demonstrates the potential of using only computer vision techniques to extract relevant information from everyday objects to create a ubiquitous computing environment, without any *a priori* preparation of the objects.

8.2 Informal User Study on ARCADE

Based on the feedback from the preliminary user study conducted for the earlier version of ARCADE, the ARCADE system has been improved. Bare hand interaction has been implemented as the main interaction tool, which makes the interaction more intuitive. The building block approach has been introduced and more sophisticate modeling operations are added. The 3D models are also functional and interactive with the introduction of the F3DM, which is generated from the FBSMM. Design simulation, verification and evaluation can also be conducted in ARCADE.

8.2.1 Design Task and Participants Profile

An informal user study has been conducted for ARCADE while it is still under development. Three participants are invited to test the system, one of whom is an 11 year old child, while the other two subjects are aged 25 and 28. This is to garner early user feedback on the system. The scope of the informal study is to find out if ARCADE is intuitive for the user to create designs and interact with them.

The user study started by showing the subjects what can be achieved in ARCADE. The building block modeling approach was easily understood by the child and was referred to virtual LEGO to him. On the other hand, modeling by extrusion from 2D sketches was quite novel to all the subjects as they did not know that 3D models can be created from extruding 2D sketches. The child was especially surprised to see that 3D models can be created in this manner. As a result, the building block modeling approach was selected as the preferred form of modeling.

After showing the capabilities of ARCADE, a simple design task of designing a toy car was given to the participant. The design of the car commenced with some training for the participant to get used to the system. Each participant carried out the design task with as little intervention as possible. The participant had to test the designed with the real objects, which included an arch that the car must pass through. The design task was completed when the design requirements were met. User feedback was gathered at the end of the task using a qualitative interview, asking about the benefits, drawbacks and recommendations for the system.

8.2.2 Results and Discussion

The benefits mentioned by the subjects are:

- The system allowed them to design a car without having to draw well. The child claimed that he would not be able to draw his desired car on paper.
- Creating 3D shapes in the system was easy and combining them to create the car is like playing with LEGO.
- The ability to move virtual things in 3D using bare hands was quite fun and interesting.

The drawbacks raised by the subjects are:

- System only supported one user and it is quite lonely to design alone.
- Time had to be taken to learn and get used to the system.

- Slight fatigue was reported after prolonged use as the hands were moving in the air.

The suggestions made by the subjects are:

- Realistic rendering of the 3D models so that the designed models will look more fun and interesting to play with.
- Draw on real objects so that he can design with both virtual and real models.

The user feedback from the informal user study was used to improve ARCADE and this leads to the final version which is presented in this thesis.

8.3 Formal User Study on ARCADE

A formal user study has been conducted for the final version of ARCADE. The purpose of the user study is to:

- Test the intuitive 3D modeling in ARCADE to determine whether it is intuitive to create 3D models and its effectiveness in externalizing the user's ideas.
- Test the interactive 3D models and understand how they can be used for conceptual design. Are they able to help a user understand the design better? Will they allow the user to explore the solution space and reflect on the design better?
- Test the design simulation, verification and evaluation provided by ARCADE. Does the simulation allow the user to understand the design better? Can the verification process help the user identify potential design

issues with the design? Does the evaluation process allow the user to select the best solution for further development?

8.3.1 Design Task and Participants Profile

The participants for the user study are divided into two groups, namely, one group with design experience (Group A) and another with little and no design experience (Group B). The design task for both groups is the same, which is the conceptual design of a robotic table-top cleaner similar to Case Study 1 presented in Section 7.2. The categorization of the different groups is to differentiate the opinions by the professional designer and the layman. There is a total of 20 participants with five in Group A and 15 in Group B. The design experience ranges from 1-5 years for the participants in Group A. The age range of the participants is from 23 to 31 years old.

Both groups have to generate the 3D models using the I3DMM of ARCADE for at least one of the design concept. In order to create the F3DMs for all the design concepts, the participants have to define the PUM for every concept that they create. ARCADE will perform the backend reasoning processes. Design verification of the functional and geometrical aspects of the design will be carried out first and the participants will be prompted to make necessary corrections to the design concepts. This will be followed by the behavior simulation and design evaluation of the design concept in terms of the user interaction with it.

8.3.2 Questionnaires and Protocols

At the end of the design task, the participants are required to fill up two questionnaires to garner feedback on ARCADE. The first questionnaire is the AttrakDiff survey (Hassenzahl, 2004), which consists of questions that measures the subjects' perceived usability and interactivity of the ARCADE system. The second questionnaire contains questions in three areas: intuitive 3D modeling, interactive 3D models and design simulation, verification and evaluation.

AttrakDiff is a questionnaire used to measure the usability and attractiveness of interactive products and systems developed by Hassenzahl (Hassenzahl, 2004). The questionnaire uses 28 pairs of opposite adjectives for the user to choose to indicate the perception of the product (see Figure 8.5). The adjective-pairs are collated to four evaluation dimensions:

1. Pragmatic Quality (PQ), which measures the product's ability to allow the user to complete the task and is generally linked to the usability.
2. Hedonic Quality – Stimulation (HQ-S), which measures the product's ability to stimulate and interest the user. It generally linked to the novelty and innovation of the product.
3. Hedonic Quality – Identity (HQ-I), which measures how the user identify with the product and express oneself with the product. It is linked to the emotional attachment one has with a product.
4. Attractiveness (ATT), which measures the general quality and beauty of the product based on the user's perception.

In general, the use of the AttrakDiff questionnaire is to evaluate the usability and general perception of ARCADE. In particular, the PQ and HQ-S are evaluation dimensions that will be studied in depth. An example of a question in Attrakdiff is shown below and the user will have to select a value between the “human” and “technical” adjectives.

human ○ ○ ○ ○ ○ ○ ○ technical

The second questionnaire will seek user feedback on the three areas of the ARCADE system, namely the intuitive 3D modeling techniques provided by the I3DMM interactive 3D models in the form of the F3DM and its creation process, and design simulation, verification and evaluation. The second questionnaire is found in Appendix A. It is used to measure the user perception on:

1. The 3D modeling techniques compared with other methods such as sketching and conventional CAD.
2. The usefulness and interactivity of the F3DM.
3. The applicability of the design verification and evaluation methods provided in ARCADE.

Most of the questions use a Likert scale to measure the user perception and some of the questions require the user to rank different items. There are a few open-ended questions that require qualitative feedback from the user.

8.3.3 Results

AttrakDiff Questionnaire

Figure 8.3 shows the general result of the user perception on ARCADE based on a matrix where the values of hedonic quality (HQ) are measure in the vertical axis and values of PQ in the horizontal axis. ARCADE scores above average for both PQ and HQ and is rated to be "rather desired" by the users.

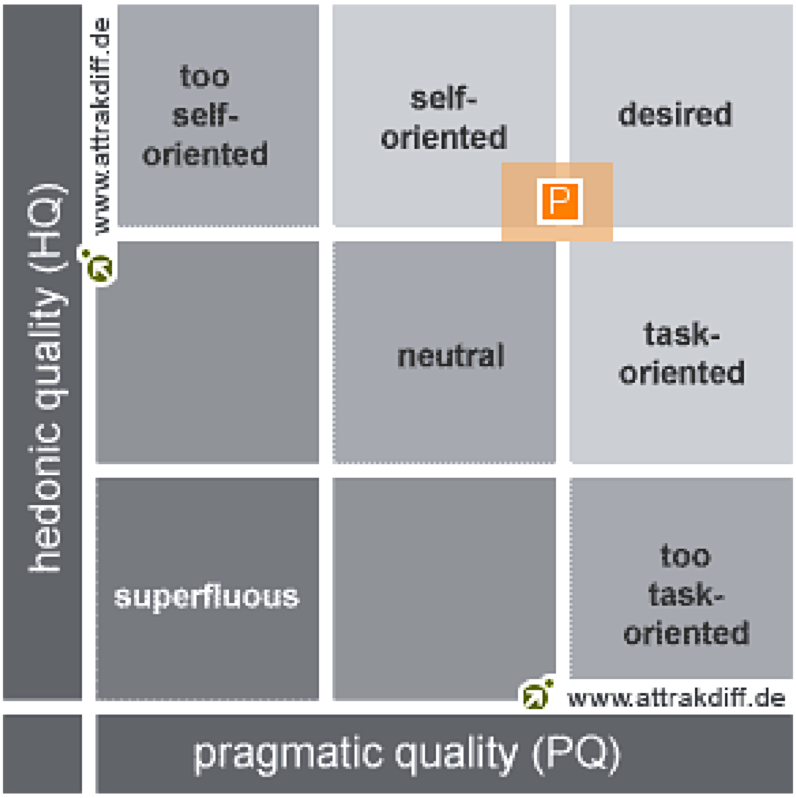


Figure 8.3: Result of the AttrakDiff questionnaire

Figure 8.4 shows the scores for the four evaluation dimensions for ARCADE. All four dimensions of ARCADE are above average. PQ and HQ-I are slightly above average and this indicates that there is some room for improvement in these areas. HQ-S is scored higher than both PQ and HQ-I and this meant that ARCADE could stimulate the users, awake their curiosity and motivate them. This is expected as AR is quite a novel concept to most of the participants. The overall attractiveness of ARCADE is also rated highly by the participants.

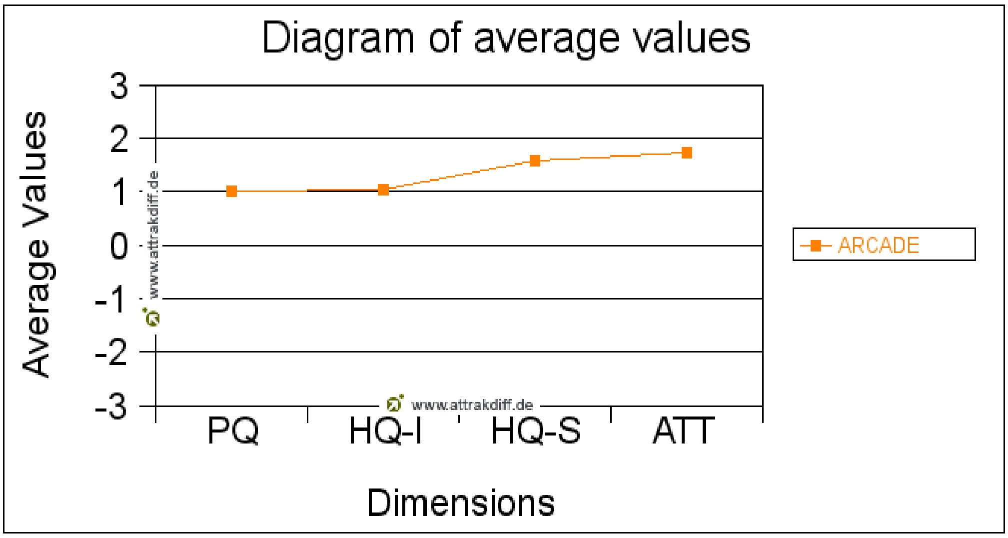


Figure 8.4: Scores of the four evaluation dimensions for ARCADE

More in-depth analysis is conducted based on the descriptions of the word-pairs shown in Figure 8.5. For PQ, ARCADE scores particularly well in being practical and manageable which implies that the participant can use it to create the design without much problem. However, it is deemed to be less human and moderately technical for the participants. For HQ-I, ARCADE scores well in most area except for one, “cheap-premium”. This is understandable as ARCADE is a research work and is less polished and sophisticated as compared to commercial 3D modeling system. HQ-S is the area where ARCADE scores very well and it is perceived to be inventive, creative, novel and innovative. Two word-pairs that ARCADE did not score quite as well on are “cautious-bold” and “undemanding-challenging”. ARCADE could be perceived to be not as bold due to the use of 3D model for design being well established. For “undemanding-challenging”, one of the goals is to make it easy for user to create 3D models in ARCADE and thus it is better for ARCADE to be perceived as undemanding compared to challenging. The average score for this word pair implies that ARCADE can still be made easier to use but this would affect its HQ-S score. For ATT, ARCADE scores moderately well for all the word-pairs with not much deviation.

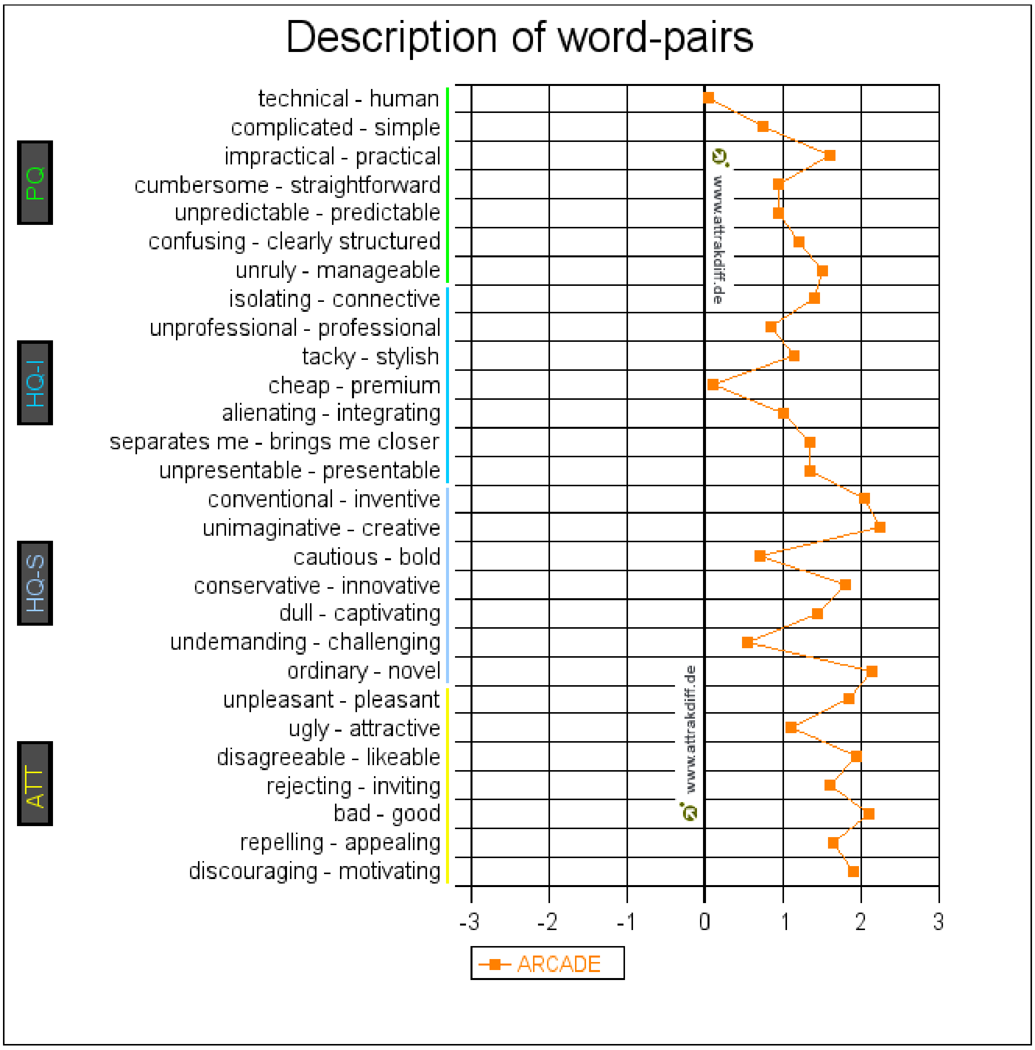


Figure 8.5: Detailed results of the word-pairs of ARCADE

Second Questionnaire

The first part of the second questionnaire addresses the participants' perception on the modeling approaches provided by ARCADE. The participants were first surveyed on their familiarity with various design methods such as building blocks, extrusion, sketching and clay modeling. Figure 8.6 shows the responses to the question. For building blocks, both Group A (4.8) and Group B (5.1) shared the same level of familiarity. For extrusion, which is commonly used in conventional CAD systems, Group A participants (5.6) had a significantly higher level of familiarity compared to Group B participants (3.6). This is expected as most of the Group B participants did not have any experience using CAD systems. For sketching, Group A participants (5.0) had a higher level of familiarity than the Group B participants (4.3) as well. However, the difference is not as significant as extrusion as sketching is more common. For clay modeling, both Group A (2.6) and Group B (2.7) participants had the same level of familiarity. This is because Group A participants do not use clay modeling for their design work and thus had little familiarity with it.

The next question queries the participants on the ease of using various techniques for creating 3D models, namely, the building block approach in ARCADE, the extrusion approach in ARCADE, sketching and conventional CAD modeling. Figure 8.7 shows the responses. In general, Group A participants found all of the methods easier to use compared to the Group B participants. For Group A participants, the extrusion approach in ARCADE was the easiest to use (6.0), followed by building block approach in ARCADE (5.6). Sketching and CAD

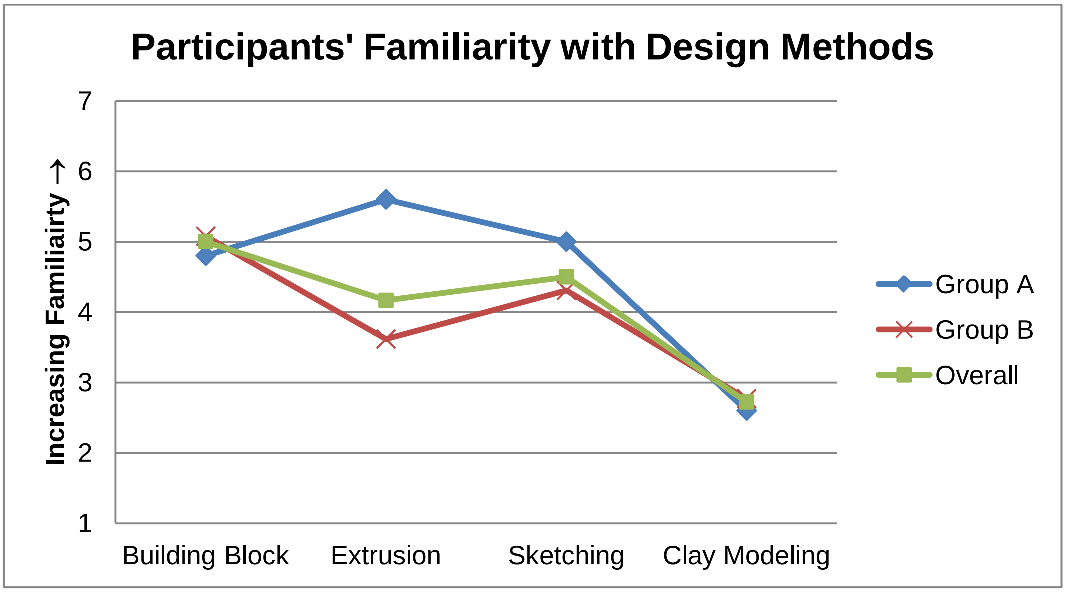


Figure 8.6: Responses to the participant's familiarity with various design methods

modeling had the same score of (5.4). As the difference in score is small, it can be deduced that Group A participants found all the methods easy to use. For Group B participants, the easiest method was the building block approach in ARCADE (4.8), followed by extrusion approach (4.2), sketching (4.1) and CAD (3.4). This implied that the building block approach is more intuitive to the laymen. The extrusion approach is also deemed to be easier than CAD and this suggested that the bare hand interaction techniques are more intuitive. The difference between Group A and Group B participants' perceived ease-of-use for the various methods is likely due to the design training that Group A participants had undergone. Methods, such as sketching and CAD modeling, that are more difficult to Group B participants were easy to them.

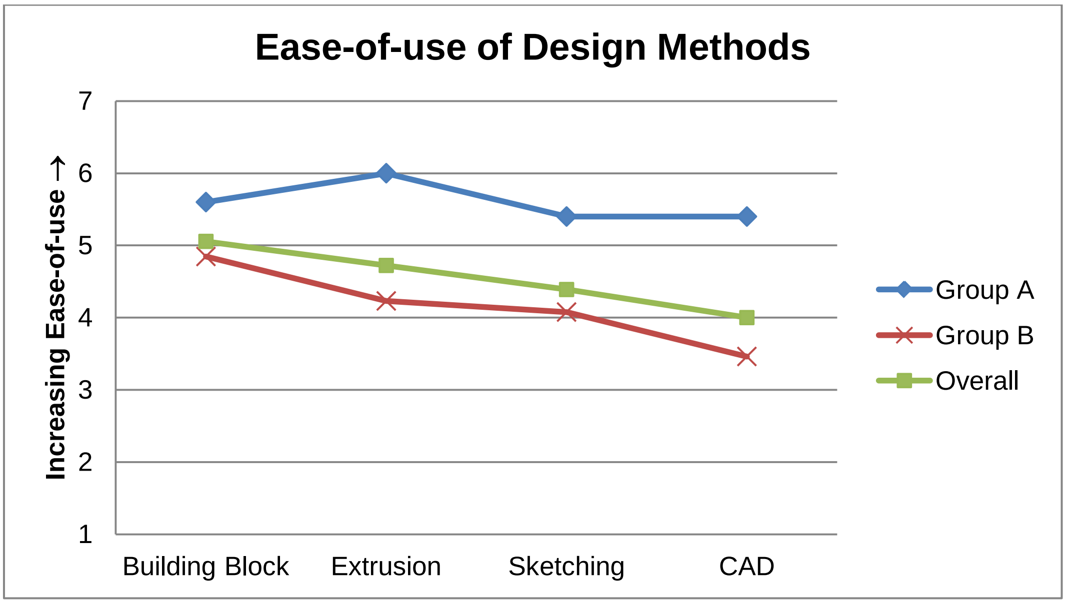


Figure 8.7: Responses to the ease-of-use of various design methods

The participants were also asked to rank their preferred methods for design among the four methods. Figure 8.8 shows the results. Group A participants ranked the building block, extrusion and sketching similarly (2.4) and the least preferred was CAD (2.8). This suggests that the modeling approaches in ARCADE are comparable to sketching and is better than CAD modeling in the views of the Group A participants, i.e. designers. However, the sample size is small to make a conclusive statement. Group B participants ranked the building block approach (1.2) as the best design methods significantly higher than other methods. Extrusion approach in ARCADE was ranked a distant second (2.6) followed by sketching (2.8) and CAD (3.3). Therefore it can be concluded that building block approach in ARCADE is the most preferred design method for Group B participants, i.e. laymen.

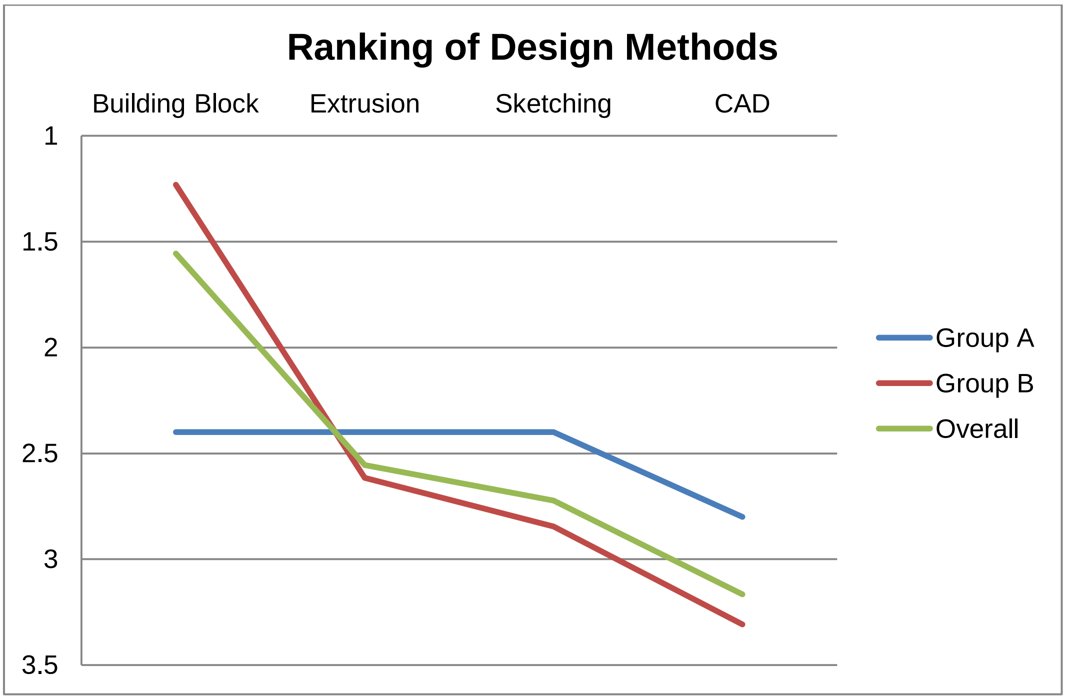


Figure 8.8: Ranking of the design methods

The second part of the questionnaire queries the participants on the F3DM. They were asked whether PUM is useful in defining the functions they need for the table-top cleaner, whether the F3DM is a realistic representation of their designs and whether F3DM is interactive when they were testing their designs. In general, all participants found PUM to be useful (Group A: 5.6; Group B: 5.4). F3DM was deemed to be realistic (Group A: 5.4; Group B: 5.1) and interactive (Group A: 6.2; Group B: 5.6). There were also little differences in the opinions between both groups of participants. Some of the Group A participants found the interactivity of the F3DM very useful and commented that it was something that conventional CAD systems are not able to support.

The third part of the questionnaire queries the participants on the design verification and evaluation aspect of ARCADE. The participants were only required to response if ARCADE or they had spotted issues with the design. As the design of the table-top cleaner is relatively simple, there were only a few reported design issues. The reported design issues include missing functions to navigate past obstacles, access hard-to-reach areas, and prevent the table-top cleaner from falling off the table. Some missing components include a bumper to protect the cleaner and a bump sensor to sense collision with objects. For functional-geometrical design issues, there is no reported issue. However, all of the 3D models created by the user had undergone correction to the alignments of wheels so that they are co-axial and other components to achieve aesthetics symmetry. This is considered as a beautification process to some of the participants rather than a design issue. For the evaluation of the hand strain when the user interact with the table-top cleaner, hand strains were detected for eight of the participants and they rate the usefulness of the hand strain analysis to be 5.0, which implied that it is rather useful for evaluating the ergonomics of the design.

The final part of the questionnaire asks the participants to comment on the ARCADE system as a whole. Some of the frequently used positive adjectives include “innovative”, “interactive”, “easy” and “intuitive”. This means that ARCADE has achieved its research objectives in the participants’ views. On the other hand, some of the frequently used negative adjectives include “jittery”, “not sleek” and “not sophisticated”. “Jittery” is attributed to the jittering when the camera is tracking the AR marker and the detection of the hands during interaction. The unpredictable movements of the hands lead to tracking loss and errors which cause the 3D models to jitter. As ARCADE is considered as a research application, it cannot be compared to commercial CAD systems in terms of sleekness and sophistication.

9. Conclusion

This chapter concludes the thesis. The research contributions of this thesis will be presented and discussed followed by the limitations of the research and recommendations for future work.

9.1 Research Contributions

The main objective of this research is to develop an AR 3D Design Space, which is named ARCADE in this thesis. Using ARCADE, conceptual design can be conducted to consider both the functional and geometrical aspects of the design concurrently. The functions of the product can be defined by the user using the PUM and the geometry is defined from the creation of 3D models in ARCADE. They will be used to create a F3DM model which can be interacted with during design evaluation. The consistency of the functional, geometrical and behavioral aspects of the design is maintained using a FBS modeling framework. In addition, ergonomics of the design can be evaluated using a hand strain detection and analysis methodology.

In the design and development of the ARCADE system, the following research contributions have been made:

- **An intuitive method for generating 3D models in an AR design environment using bare hand interaction has been developed.**

3D model generation in ARCADE uses two approaches that are both familiar and easy-to-use for the user: Building Block approach and Extrusion approach. They are relatively easier to use compared to conventional CAD and are comparable to 2D sketching for generating design ideas from the results of the user studies conducted. ARCADE provides direct manipulation of the 3D models using the hands to create new design. This allows better exploration of the design space. The user can “think with the hands” to generate ideas for the product. Editing functions are also provided for the user to modify the 3D models directly in ARCADE. Design innovation can be improved as the designer has less cognition load on the creation of 3D models and can focus more on generating different ideas. In addition, the designers can derive new ideas by modifying and playing with existing 3D models. The 3D models can be used directly on conventional 3D CAD software for the later stages of the design process such as detailed design. This integration increases the efficiency of implementing a design concept from the conceptual design stage to the final product stage.

- **Functional 3D models (F3DM) are introduced to represent the user’s conceptual design and the Function-Behavior-Structure (FBS) modeling framework that synthesizes the function model, behavior model and 3D model to form the F3DM has been developed.**

F3DM are able to reflect the functional behavior of the design on top of its geometry. It is derived from the PUM and 3D models that are defined by the users for the conceptual design. It can represent the product holistically in terms of its functions, behavior and product structure. The user can interact with the F3DM during design evaluation and learn more about the design compared to just a 3D model representation of the design. To ensure the consistency of the PUM and 3D model and synthesize the various models to represent the product, the FBS modeling framework has been developed. It uses a FBS modeling language to represent the product and has various reasoning algorithms and methods to obtain the function, behavior and product structure from the user-defined PUM and 3D model. The F3DM is interactive and can be used as a realistic prototype during conceptual design. It can be used to improve the understanding of the product, communicate the design to others and reflect upon by the user. The interactivity of the F3DM is an advantage that 3D model holds over 2D sketching. While 2D sketching may still be the dominant tool for idea generation, the interactivity of the F3DM will encourage more users to use 3D models. From the user studies, the participants found the creation of the F3DM by defining the PUM to represent the product's functions and generating the 3D models to be useful. They also indicated the F3DM is a realistic and interactive representation of their design.

- **A design simulation system that allows the F3DM to behave functionally in the same manner as the actual product for design evaluation has been developed.**

From the behavior model in F3DM of the product, the functional behavior of the F3DM can be simulated when the user interacts with it during design evaluation. The ARCADE system can track the user interaction, determine the resultant behavior from the behavior model of the F3DM and modify the geometrical parameters of the F3DM to simulate the behavior. The behavior simulation of the F3DM helps the user to understand the product better by actually using the product. The use of the product is hard to quantify during conceptual design. By having a functional prototype that behaves like the final product, the usability can be studied and be used as a criterion for design evaluation.

- **A design verification mechanism that ensures the consistency between the functional and geometrical aspects of the F3DM has been developed.**

The function model and product structure model of the F3DM can be checked for consistency with the design verification mechanism. Critical components that are required for the design can be identified and the user will be prompted to create the 3D models. In addition, the geometrical relationships between components are verified with their functional relationships to ensure that they are able to perform their functions. This can assist the user to identify design issues that may have been overlook when the focus is on either the form or function of the product. The design concept created using ARCADE will always be functionally and geometrically feasible.

- **A hand strain detection methodology to evaluate the ergonomics of the handling of the 3D models for design evaluation has been developed.**

ARCADE is able to track the user's hands during the handling of the product and identify hand strain incidents. During design evaluation, the user can use and handle the product and this is a more practical way of evaluating the product. Hand strain incidents during the handling of the product indicate that the design has ergonomic issues that need to be addressed. This allows the user to evaluate the usability and ergonomics issues involving the handling of the product can be captured as an evaluation criterion. This will lead to the selected concept being user-friendly and has fewer issues with the ergonomics. To the best of the author's knowledge, this is the first system that utilizes hand strain detection for design evaluation in an AR design environment.

9.2 Limitations and Recommendations for Future Work

The limitations of the research and recommendations for future work are listed as follows:

- The intuitive 3D modeling module can be more intuitive with the introduction of other intuitive 3D modeling methods such as direct clay modeling with the hands or 3D sketching. Clay modeling has been experimented for this research and it has potential to improve the intuitiveness of 3D modeling. However, the computational cost is high and often leads to non-real-time modeling. During clay modeling, it is also difficult to capture the user intentions on how the shape should be modified. With the advances of computational power of newer processors, it may be possible to implement clay modeling in the future for I3DMM.
- The reasoning processes to generate the F3DM from the 3D models and PUM are highly dependent on the product database. The database can be expanded to increase the types of products that can be designed. However, it is nearly impossible to develop generic processes that can be used to design any products. Evolutionary algorithms may be implemented to develop new *FBS_Primitives* from existing ones in the future to increase products that can be designed using ARCADE.

- Design verification in ARCADE only addresses the functional and geometrical aspects of the design. Other aspects can be added in the future. However this will require additional models to be implemented, which may increase the complexity of the reasoning processes.
- The design evaluation on the ergonomics is currently limited to the hand strains for the hand region above the wrist. The whole upper extremity can be considered for future implementation. However, this will require a much larger tracking space, leading to higher computational and setup cost and decrease the portability of the system.

9.3 Conclusion

AR is currently one of the fastest growing technologies in the world. It has many potential uses in various fields, and one research area that has been identified in this project is conceptual design. The technological capabilities of AR and the requirements for conceptual design are highly compatible. The goal of this research integrates them to create a design system – ARCADE and the research and development of the system is documented in this thesis.

Publications from this Research

Ng, L.X., Wang, Z.B., Ong, S.K., and Nee, A.Y.C. (2013). Integrated product design and assembly planning in an augmented reality environment. *Assembly Automation.* 33(4). pp. 345-359.

Wang, Z.B., Ng, L.X., Ong, S.K. and Nee, A.Y.C. (2013). Assembly Planning and Evaluation in an Augmented Reality Environment. *International Journal of Production Research*. 50(23-24). Pp. 7388 - 7404.

Ng, L.X., Ong, S.K., and Nee, A.Y.C. (2013), Conceptual Design using Functional 3D Models in Augmented Reality. *International Journal of Interactive Design and Manufacturing*, under revision December 2013.

Ng, L.X., Ong, S.K., and Nee, A.Y.C. (2010). ARCADE: a simple and fast augmented reality computer-aided design environment using everyday objects. *IADIS Interfaces and Human Computer Interaction 2010 (IHCI 2010) Conference*, 28-30 July, Freiburg, Germany, pp. 227-234.

Related Paper

Ng, L.X., Oon, S.W., Ong, S.K. and Nee, A.Y.C. (2011). GARDE: a gesture-based augmented reality design evaluation system. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. 5(2), pp. 85-94.

References

20-Sim. (2010). Controllab Products B.V. Retrieved from: http://www.20sim.com/ (last accessed 21 Feb 2013).

Aoyama, H. & Kimishiman, Y. (2009). Mixed reality system for evaluating designability and operability of information appliances. *International Journal on Interactive Design and Manufacturing*, 3(3), pp. 157-164.

ARToolkit. (2007). Retrieved from: http://www.hitl.washington.edu/artoolkit/ (last accessed 12 February 2012).

Azuma, R.T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*. 6(4), pp. 355-385.

Barfield, W., Shieldst, R. & Cooper, S. (1993). A survey of computer-aided design: Implications for creativity, productivity, decision making, and job satisfaction. *International Journal of Human Factors in Manufacturing*, *3*(2), 153-167.

Bracewell, R.H. and Sharpe, J.E.E. (1996). Functional descriptions used in computer support for qualitative scheme generation – Schemebuilder. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing.* 10(4), pp. 333-346.

Branit, B. (2007). World Builder. Retrieved from: http://vimeo.com/3365942 (last accessed 20 Dec 2012).

Brown, T. (2009). Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. HarperCollins Publishers, New York, USA.

Chaffin, B.D., Andersson, G.B.J & Martin, B.J. (2006). *Occupational biomechanics*. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 4th ed.

Chakrabarti, A. and Bligh, T.P. (2001). A scheme for functional reasoning in conceptual design. *Design Studies.* 22(6), pp. 493-517.

Chandrasekaran, B. (2005). Representing function: relating functional representation and functional modeling research streams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing.* 19(2), pp. 65-74.

Chandrasekaran, B. and Josephson, J.R. (2000). Function in device representation, *Engineering with Computers.* 16(3), pp. 162-177.

Chen, Q., Rahman, A.M., Shen, X.J., El Saddik, A. and Georganas, N.D. (2007). Navigating a 3D virtual environment of learning objects by hand gestures. *International Journal of Advanced Media and Communication*. 1(4), pp. 351-368.

Deng, Y.M. (2002). Function and behavior representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing.* 16(5), pp. 342-362.

Dhawale, P., Masoodian, M. and Rogers, B. (2006). Bare-hand 3D gesture input to interactive systems. *Proceedings of the 7th Int. Conf. NZ Chapter of the ACM's SIG on Human-Computer Interaction, Christchurch, New Zealand, 6-7 July 2006.* pp. 25-32.

Do, T. V. & Lee, J. W. (2008). 3DARModeler: A 3D Modeling System in Augmented Reality Environment. *International Journal of Computer Systems Science and Engineering*. 4(2), pp. 145-154.

Du, H. and Charbon, E. (2007). 3D Hand Model Fitting for Virtual Keyboard System. *Proceedings of IEEE Workshop on Applications of Computer Vision, Austin, USA, February 2007*. pp. 31.

Duh, H. B., & Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, Cambridge, United Kingdom, 15-18 September 2008*. pp. 193-202.

Feix, T., Pawlik, R., Schmiedmayer, H., Romero, J., & Kragic, D. (2009). A comprehensive grasp taxonomy, *In Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*. pp. 2-3.

Fiorentino, M., de Amicis, R., Monno, G., & Stork, A. (2002). Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design. *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '02), Darmstadt, Germany, 30 September-1 October 2002*. pp. 86-94

Fiorentino, M., Monno, G., & Uva, A. E. (2009). Tangible digital master for product lifecycle management in augmented reality. *International Journal on Interactive Design and Manufacturing,* 3(2), pp. 121-129.

Fuge, M., Yumer, M.E., Orbay, G. and Kara, L.B. (2012). Conceptual design and modification of freeform surfaces using dual shape representations in augmented reality environments. *Computer-Aided Design.* 44(10), pp. 1020-1032.

Gero, J.S, and Kannengiesser, U. (2004). The situated function-behavior-structure framework. *Design Studies,* 25(4), pp. 373-391.

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

Hamer, H., Schindler, K., Koller-Meier, E. and Van Gool, L. (2009). Tracking a hand manipulating an object. *Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September-2 October 2009.* pp. 1475-1482.

Hassenzahl M. (2004).The interplay of beauty, goodness and usability in interactive products. *Human–Computer Interaction.* 19(4), pp. 319–349.

Hilliges, O., Izadi, S., Wilson, A. D., Hodges, S., Garcia-Mendoza, A., and Butz, A. (2009). Interactions in the air: adding further depth to interactive tabletops. *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, Victoria, Canada, 4-7 October 2009.* pp. 139-148.

Hirtz, J., Stone, R., McAdams, D., Szykman, S. and Wood, K. (2000). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design.* 13(2), pp. 359-370.

Hong, D., H, T., Haller, M., Takemura, H., Cheok, A. D., Kim, G. J., et al. (2008). Advances in Tangible Interaction and Ubiquitous Virtual Reality. *IEEE Pervasive Computing*. *7*(2), pp. 90-96.

Hsu, W. and Woon, I.M.Y. (1998). Current research in conceptual design of mechanical products. *Computer-Aided Design.* 30(5), pp. 377-389.

Imrhan, S.N & Rahman, R. (1995). The effects of pinch width on pinch strengths of adult males using realistic pinch-handle coupling. *International Journal of Industrial Ergonomics*. 16(2), pp. 123-134.

Ingrassia, T. & Cappello, F. (2008). VirDe: a new virtual reality design approach. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. *3*(1), pp. 1-11.

Irawati, S., Ahn, S., Kim, J., & Ko, H. (2008). VARU Framework: Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications. *2008 IEEE Virtual Reality Conference*, *Nevada, USA, 8-12 March 2008*. pp. 201-208.

Kaufmann, H., Schmalstieg, D. & Wagner, M. (2000). Construct3D: a virtual reality application for mathematics and geometry education. *Education and Information Technologies*. 5(4), pp. 263–276.

Kelley, T. (2001). The Art of Innovation: Lessons in Creativity from IDEO, America’s Leading Design Firm. Currency-Doubleday, New York, USA.

Khan, A.A., O’Sullivan, L. & Gallwey, T.J. (2010). Effect on discomfort of frequency of wrist exertions combined with wrist articulations and forearm rotation. *International Journal of Industrial Ergonomics*. 40(5), pp. 492-503.

Kim, H. & Fellner, D.W. (2004). Interaction with hand gesture for a back-projection wall. *Proceedings of Computer Graphics International, Crete, Greece, 19 June 2004*. pp. 395-402.

Kitamura, Y., Kashiwase, M., Fuse, M., and Mizoguchi, R. (2004). Deployment of ontological framework of functional design knowledge. *Advanced Engineering Informatics*. 18(2), pp. 115-127.

Klinker, G., Dutoit, A.H., & Bauer, M. (2002). Fata Morgana – A Presentation System for Product Design, *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '02), Darmstadt, Germany, 30 September-1 October 2002*. pp. 1-10.

LAYAR. (n.d.) Retrieved from: http://www.layar.com/ (last accessed 12 Dec 2012).

Lee, T., & Höllerer, T. (2007). Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. *Proceedings of IEEE International Symposium on Wearable Computers (ISWC), Boston, USA*, *11-13 October 2007*. pp. 83-90.

Lin, M.C., Otaduy, M.A. & Boulic, R. (2008). Virtual Reality Software and Technology, *IEEE Computer Graphics and Applications,* pp. 18-19

Lipson, H. & Shpitalni, M. (2000). Conceptual design and analysis by sketching. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 14(5), pp. 391-401.

Milgram, P., Takemura, H., Utsumi, A. & Kishino, F. (1994). Augmented Reality: A class of displays on the reality-virtuality continuum. *Proceedings of*

Telemanipulator and Telepresence Technologies, Boston, USA. 31 October-1 November 1994. pp. 2351–34.

Miller, W.L & Morris, L. (1998). Fourth generation R&D. New York: Wiley.

Mistry, P., Maes, P. and Chang, L. (2009). WUW - wear Ur world: a wearable gestural interface. *Proceedings of CHI '09 Extended Abstracts on Human Factors in Computing Systems, Boston, USA, 4-9 April 2009.* pp. 4111-4116.

Modelica. (2010). Modelica Association. Retrieved from: https://www.modelica.org/ (last accessed 21 Feb 2013).

Moore, J.S. & Vos, G.A. (2004). The Strain Index, *Handbook of Human Factors and Ergonomics Methods* (edited by Stanton, N., Hedge, A., Brookhuis, K., Eduardo Salas, E. & Hendrick, H.) CRC Press. pp. 9-1 – 9-5.

Na, S., Billinghurst, M., & Woo, W. (2008). TMAR: Extension of a Tabletop Interface Using Mobile Augmented Reality. *Transactions on Edutainment I*. pp. 96-106.

Nee A.Y.C., Ong S.K., Chryssolouris G., Mourtzis D. (2012). Augmented reality applications in design and manufacturing. *CIRP Annals - Manufacturing Technology*. 61(2). pp. 657-679.

Ng, L.X., Wang, Z.B., Ong, S.K., and Nee, A.Y.C. (2013). Integrated product design and assembly planning in an augmented reality environment. *Assembly Automation.* 33(4). pp. 345-359.

Nielsen, M., Störring, M., Moeslund, T.B. and Granum, E. (2004). A procedure for developing intuitive and ergonomic gesture interfaces for HCI. *Lecture Notes in Artificial Intelligence*. pp. 409-420

Oh, J., Stuerzlinger, W. & Danahy, J. (2006). Sesame: towards better 3d conceptual design systems. In *Proceedings of the 6th conference on Designing Interactive systems, Pennsylvania, USA, 26-28 June 2006*. pp. 80–89.

OpenCSG. (2010). Retrieved from http://www.opencsg.org/ (last accessed 12 February 2013).

OpenCV. (2012), available at: http://opencv.willowgarage.com/wiki/ (last accessed 12 February 2013).

OpenGL. (1997). Retrieved from http://www.opengl.org/ (last accessed 12 February 2013).

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., Wallace, Ken; Blessing, Lucienne T.M. (Eds.). (2007). Engineering Design: A Systematic Approach. 3rd ed. Springer.

Park, H., Moon, H. C. & Lee, J. Y. (2009). Tangible augmented prototyping of digital handheld products. *Computers in Industry*. 60(2), pp. 114-125.

Park, J. (2008). Augmented Reality based Re-formable Mock-up for Design Evaluation. *International Symposium on Ubiquitous Virtual Reality, Gwangju, Korea, 10-13 July 2008*. pp. 17-20.

Park, J. Y. & Lee, J. W. (2004). Tangible Augmented Reality Modeling. *Entertainment Computing - International Conference on Entertainment Computing 2004, Eindhoven, Netherlands, 1-3 September 2004*. pp. 254-259.

Piekarski, W. & Thomas, B.H. (2003). Tinmith-Mobile Outdoor Augmented Reality Modeling Demonstration. *Proceedings of the 2nd IEEE/ACM*

International Symposium on Mixed and Augmented Reality, Tokyo, Japan, 8-10 October 2003. pp. 317.

Protégé. (2012). Retrieved from http://protege.stanford.edu/ (last accessed 07 Apr 2013).

Pugh, S. & Clausing, D. (1996). Creating Innovative Products Using Total Design: the Living Legacy of Stuart Pugh. 1st. Addison-Wesley Longman Publishing Co., Inc.

Robertson, B. & Radcliffe, D. (2009). Impact of CAD tools on creative problem solving in engineering design. *Computer-Aided Design*. 41(3), pp. 136-146.

Ronald Sidharta, J. O. (2006). Augmented Reality Tangible Interface for Distributed Design Review. *International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06), Sydney, Australia, 26-28 July 2006*. pp. 464-470.

Schlattmann, M., Broekelschen, J. and Klein, R. (2009). Real-time are-hands-tracking for 3D Games. *Proceedings of IADIS International Conference Game and Entertainment Technologies*, pp. 59-66.

Segen, J. & Kumar, S. (1998). Gesture VR: Vision-based 3D hand interface for spatial interaction. *Proceedings of the sixth ACM international conference on Multimedia, Bristol, United Kingdom, 13-16 September 1998*. pp. 455–464.

Sharp, H., Y. Rogers, and J. Preece (2007). *Interaction Design: Beyond Human-Computer Interaction* (2 ed.). Wiley.

Shen, Y., Ong, S.K., & Nee, A.Y.C. (2008). Product information visualization and augmentation in collaborative design. *Computer-Aided Design*. 40(9), pp. 963-974.

Shin, M., Kim, B., and Park, J. (2005). AR Storyboard: An Augmented Reality Based Interactive Storyboard Authoring Tool. *Proceedings of the 4th IEEE/ACM international Symposium on Mixed and Augmented Reality, Vienna, Austria, 5-8 October 2005*. pp. 198-199.

SolidWorks API. (2012).*SolidWorks Corporation*. Retrieved from http://www.solidworks.com/sw/support/api-support.htm (last accessed 12 Feb 2013).

Suh, N. P. (2005). *Complexity: Theory And Applications (Mit-Pappalardo Series in Mechanical Engineering)*. Oxford University Press.

The New Oxford American Dictionary. (2005). New York: Oxford University Press.

Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, Ch. & Kimura, F. (2009). Design methodologies: Industrial and educational applications, *CIRP Annals - Manufacturing Technology*. 58(2), pp 543-565

TRIZ. (n.d.) Retrieved from http://www.triz-journal.com/ (last accessed 12 Feb 2012).

Ullman, D.G. (2009) The Mechanical Design Process, 4th edition, McGraw-Hill Higher Education.

Ulrich, K.T. & Eppinger, S.D. (2004) *Product Design and Development*, 3rd Edition, McGraw-Hill, New York.

Umeda, Y., Ishii, M., Yoshioka, M. and Tomiyama, T. (1996). Supporting conceptual design based on the function-behavior-state modeler, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing.* 10(4), pp. 275-288.

V-Collide. (1998). Retrieved from http://gamma.cs.unc.edu/V-COLLIDE/ (last accessed 12 Feb 2012).

Verlinden, J., Horvath, I., Edelenbos, E. (2006). Treatise of Technologies for Interactive Augmented Prototyping. *Proceedings of Tools and Methods of Competitive Engineering, Ljubljana, Slovenia, 18-22 April 2006*. pp. 523-536.

Vernat, Y., Nadeau, J. & Sébastian, P. (2009). Formalization and qualification of models adapted to preliminary design. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. 4(1), pp. 11-24.

Wang, Z.B. (2013). Application of augmented reality in manual assembly design and planning. Thesis dissertation.

Webel, S., Becker, M., Stricker, D. & Wuest, H. (2007). Identifying differences between CAD and physical mock-ups using AR. *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13-16 November 2007*. pp. 1-2.

Weidlich, D., Cser, L., Polzin, T., Cristiano, D. & Zickner, H. (2009). Virtual reality approaches for immersive design. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. 3(2), pp. 103-108.

Weidlich, D., Scherer, S. & Wabner, M. (2008). Analyses Using VR/AR Visualization. *Computer Graphics and Applications*. 28(5), pp. 84-86.

Wilson, P.N., Foreman N. & Stanton D. (1997). Virtual reality, disability and rehabilitation. *Disability and Rehabilitation*. 19(6), pp. 213-20.

Xin, M., Sharlin, E. & Sousa, M. C. (2008). Napkin sketch: handheld mixed reality 3D sketching. *Proceedings of the 2008 ACM symposium on Virtual reality software and technology, Bordeaux, France, 27-29 October 2008*. pp. 223-226.

Yoon, J.H., Park, J.S. and Sung, M.Y. (2006). Vision-based bare-hand gesture interface for interactive augmented reality applications. *Lecture Notes in Computer Science*, pp. 386-389

Yoshioka, M., Umeda, Y., Takeda, H., Shimomura, Y., Nomaguchi, Y. and Tomiyama, T. (2004). Physical concept ontology for the knowledge intensive engineering framework. *Advanced Engineering Informatics.* 18(2), pp. 95-113.

Zeng, Y., Pardasani, A., Dickinson, J., Li, Z., Antunes, H., Gupta, V. & Baulier, D. (2004). Mathematical foundation for modeling conceptual design sketches. *Journal of Computing and Information Science in Engineering*. 4(2), pp. 150-159.

Appendix A

Questionnaire for user study

3D Modeling

1. Rate your familiarity with the following design and construction methods, with 1 being the least familiar and 7 being the most familiar.

	Least familiar						Most familiar
Building Blocks	1	2	3	4	5	6	7
Extrusion (CAD)	1	2	3	4	5	6	7
Sketching	1	2	3	4	5	6	7
Clay modeling / Sculpting	1	2	3	4	5	6	7
Others: (please state)	1	2	3	4	5	6	7

2. Rate the ease of using the following methods for creating designs, with 1 being the most difficult and 7 being the easiest.

	Difficult						Easy
Building Blocks in ARCADE	1	2	3	4	5	6	7
Extrusion in ARCADE	1	2	3	4	5	6	7
Sketching	1	2	3	4	5	6	7
Conventional CAD	1	2	3	4	5	6	7
Others: (please state)	1	2	3	4	5	6	7

3. Rank the following methods for creating designs based on your preference, with 1 being the most preferred and 4 being the least preferred.

Building Blocks in ARCADE	________
Extrusion in ARCADE	________
Sketching	________
Conventional CAD	________

Interactive 3D Model

4. Rate the applicability of using Product Use Model (PUM) to represent the functions of your design, with 1 being not applicable and 7 being highly applicable.

	Not applicable					Highly applicable	
PUM as functions	1	2	3	4	5	6	7

5. Rate the realism of the Functional 3D model (F3DM) created to represent your design, with 1 being very unrealistic and 7 being very realistic.

	Very unrealistic						Very realistic
Realism of F3DM	1	2	3	4	5	6	7

6. Rate the interactivity of the Functional 3D model (F3DM) created to represent your design, with 1 being not interactive and 7 being very interactive.

	Not interactive					Very interactive	
Interactivity of F3DM	1	2	3	4	5	6	7

Design Evaluation

7. Did you spot any design issues when interacting with the F3DM in ARCADE?

☐ Yes ☐ No

If yes, please state the number of occurrences and briefly describe at least one of them.

8. Did ARCADE identify any functional-geometrical design issues for your design?

☐ Yes ☐ No

If yes, briefly describe the identified issues.

9. Did ARCADE detect any hand strain when you are interacting with the F3DM of your design?

☐ Yes ☐ No

If yes, rate the relevance of hand strain to the evaluation of your design, with 1 being not relevant and 7 being highly relevant.

	Not relevant						Highly relevant
Relevance of hand strain	1	2	3	4	5	6	7

10. Select and rank the following criteria that you have used to evaluate your design.

☐ Functionality ☐ Feasibility

☐ Aesthetics ☐ Usability

☐ Cost

☐ Ergonomics ☐ Others (please state): ___________

11. Please comment on the ARCADE system using adjectives if possible.

User Profile

Age:

Design experience (in years):