ABSTRACT

This paper presents an assembly model process that fully characterizes the structural and flow interactions between artifacts in a product. Reverse engineering techniques were employed during the analysis of thirty-three existing consumer products to arrive at a concise standardization of the modeling process. During the product investigation, four different types of structural interactions were identified. These structural interactions, couple, secure, position and guide, were defined using a standardized vocabulary of functional terms. These four structural interactions are rigorously described in this paper in an effort to outline an assembly model method that is accurate and repeatable. Additionally, flow interactions between components are also characterized within the presented modeling technique. A rough representation of the artifact configuration of a product can also be achieved through placement of the component structures in the model. Analysis of the consumer product set also revealed that a new design tool can be generated using the structural interaction information contained in the described assembly models.

Keywords:

1. INTRODUCTION

Conceptual design is the most important phase in any product design industry. According to Lotter [1986] up to 75% of the cost of a product is determined during the design phase. Many research efforts have focused on refining this critical phase of design to make the process more repeatable and consistently successful for both experienced and inexperienced designers. Both function-based (e.g., [Pahl and Beitz, 1988]) and catalog-based (e.g., [Ward and Seering, 1989]) methodologies have been proposed to support concept generation. In function-based approaches, the problem can be broken down to the lowest level of abstraction by deriving a functional model during the conceptual design stage. Then, various concept generation tools can be employed to arrive at the possible solutions for a given design problem [Pahl and Beitz, 1988, Hundal, 1990, Strawbridge, et al., 2004, Bryant, et al., 2005]. However, as the design problem becomes complex, application of conventional techniques becomes difficult.

Available design tools have their own advantages and disadvantages, and uniquely accommodate the different perspectives that designers have towards solving a design problem. For instance, CAD-based systems typically concentrate on capturing geometric attributes of a design instead of providing support for concept generation. Even though researchers have developed both conventional and non-conventional techniques for concept generation, a generalized computer algorithm has yet to be fully established, in part because the level of abstraction is not uniform across all methods. The solution set for a particular design problem is frequently so large that it is difficult to reduce the solutions down to a manageable number of viable results without a systematic method in place. Often, results found in the unfiltered solution set are impractical due to incompatibilities between components, either flow-wise, structurally, or both. Previous research efforts have focused on the interactions between components, but none are defined in a manner that efficiently translates into a useful form for use in concept generation [Pimmler and Eppinger, 1994; Chakrabarti and Bligh, 1994; Bracewell and Sharpe, 1996; Lee and Gossard, 1985; Callahan and Heisserman, 1997].

Artifacts within a product have two types of functionality
associated with them: conceptual functionality, and supporting functionality [Bohm, et al., 2003]. Artifacts capable of directly solving the functions described by the product’s functional model have conceptual functionality. However, some artifacts, such as a threaded fastener, have supporting functionality in a product, either in lieu of or in addition to conceptual functionality, which does not directly solve any conceptual function but is still required for the overall functionality of the product to be successful. Therefore, to effectively support conceptual design activity, adequate design knowledge, including supporting functionality, needs to be available in order to generate numerous feasible solutions. The main objective of this work is to develop a model that captures the supporting functionality of a product, increasing the design knowledge available to effectively support automated conceptual design. The research explores the representation of supporting functionality through a study of products with diverse functionality, varying part count, and different energy domains.

2. BACKGROUND

We begin with a review of the state of the art in area of conceptual design research and areas that support automated concept generation. In particular, we first review systematic approaches to conceptual design and examine previous research efforts focused on the exploration of different methods for representing component interactions. Finally, we focus on existing product function representation and design knowledge collection to support concept generation.

2-1. Related Research into Concept Generation and Component Interaction Representation

Several researchers have employed different techniques to tackle the problem of automating the concept generation phase of design. Hundal [1990] designed a program for automated conceptual design that associates a database of solutions for each function in a function database. The user inputs the functions into a function structure in order to generate functional variants. The accuracy of a returned solution is dependent upon the designer’s ability to break down the black box function of a particular problem to appropriate sub-functions and input/output quantities. The software is limited to generating solutions to functions contained in the available functional database.

Ward and Seering [1989] developed a mechanical design “compiler” to support catalog-based design. Built up from a database of “basic sets” of artifacts represented by a catalog number, the system takes appropriate schematics, specifications, and desired functionality for a mechanical design as inputs and returns the catalog numbers for an optimal solution. The compiler eliminates catalog numbers that are incompatible with the previously implemented components.

Bracewell and Sharpe [1996] developed “Schemebuilder,” a software tool using bond graph methodology to support the functional design of dynamic systems with different energy domains. The software utilizes a predefined functional embodiment knowledge base to seek solutions to the conceptual design functions. The system incorporates bond graph decomposition rules, a functional-embodiment database, and a component database and defines ports to characterize each component and limit connections to compatible energies. The designer enters the component function through FESTER (Functional Embodiment Structure—Extended Recursively), a function–means tree that supports hierarchical links that represent a process of function embodiment. However, since the system implements bond graph decomposition techniques, it cannot efficiently deal with function not represented as power flow.

Strawbridge, et al. [2004] developed a concept generation technique that utilizes a function-component matrix and a filter matrix to generate a morphological matrix of solutions for functions in a conceptual functional model. The function-component matrix, \( \chi \), uses columns of components and rows of functions to characterize component functionality. Cell values for \( \chi \) are either zero or non-zero depending on whether component \( j \) solves function \( i \). Aggregate \( \chi \) matrices can be constructed from individual product function-component matrices and filtered to generate a matrix describing the complete solution set.

Although useful, the methods described above for concept generation are not complete in the sense that component-component compatibility (both flow related and structurally) is not fully accounted for and entire product assembly solutions are not generated. Flow incompatibility is defined as the inability of an artifact to connect with another artifact due to differences in energy domains, e.g. a device whose output flow is mechanical energy cannot be directly connected to an artifact with an electrical energy input flow. Structural incompatibility is defined as the inability of connecting artifacts to be directly assembled together.

Several research efforts have focused on representing the interactions between artifacts within a product. Pimmler and Eppinger [1994] describe the interactions that occur between the elements of the product in terms of spatial, energy, information, and materials. Interactions between components are ranked on a scale (-2, 2), where -2 identifies that the interaction is detrimental for the desired product functionality and +2 identifies that the interaction is required for proper function of the product. The interactions between components are then used to define product architecture. However, the information modeled is not sufficient to be used at the concept generation stage. The previously described “Schemebuilder” software by Bracewell and Sharpe [1996] uses a bond graph methodology that utilizes ports to characterize each component and only compatible energy ports can be connected to each other. The Schemebuilder can support functional design of dynamic systems with different energy domains.

Chakrabarti and Bligh [1994] model the design problem as a set of input–output transformations. Structural solutions to each of the instantaneous transformation are found, and infeasible solutions are filtered according to a set of temporal reasoning rules. The temporal reasoning rules are only of a dynamic nature, i.e. they deal with force and torque issues, and are used to filter out the infeasible solutions. Lee and Gossard [1985] developed a data structure for representing assemblies in a hierarchical manner. The relationships between the components in an assembly are represented by means of “virtual link”—a set of information required to completely describe the relationship. Callahan and Heiserman [1997] also developed a hierarchical assembly representation and unique naming scheme that combines associated design concepts and supports design reuse. The system supports
effective management of design and manufacturing data for complex products. Some of the methods described above capture the interactions between components at the geometric level [Lee and Gossard, 1985; Callahan and Heisserman, 1997] while others [Bracewell and Sharpe, 1996; Chakrabarti and Bligh, 1994] capture energy level interactions between artifacts. In this paper, we describe a modeling representation that captures the complete structural and flow compatibility of existing product artifacts by a reverse engineering method. By having accurate and complete assembly design knowledge available, more powerful tools for automatically creating multiple conceptual variants of complete products can be crafted.

2-2. Functional Basis

Intending to span the entire mechanical design space without repetition, Hirtz, et al. [2002] developed a uniform functional language to support functional modeling known as the Functional Basis. The Functional Basis defines a comprehensive set of functions and flows at three levels of abstraction to support both high and low level functional modeling independent of the physical structure of the artifact. The Functional Basis is useful for modeling engineering artifacts and systems with different energy domains. Designers can choose at which level of abstraction to define a conceptual model using the set of equivalent functions and flows at different stages of product design. For instance, designers at the beginning stages of product design may define the general functionality of the product using the most general primary level of detail, while designers in the concept generation phase may find function definitions more useful defined at the tertiary level. Ultimately, the Functional Basis presents a common platform for designers to uniformly represent any design problem without ambiguity.

2-3. Design Repository

At the University of Missouri–Rolla, Bohm, et al. [2004] used the Functional Basis to develop a design repository that stores design knowledge of artifacts from different energy domains. The repository contains information about each artifact’s function, associated flows, physical characteristics, and potential failure modes. Additionally, the repository offers design tools such as matrices of function-component relationships (function-component matrices or FCMs) and component-component relationships (Design Structure Matrices or DSMs) to aid designers. FCMs use rows of functions and columns of components to convey component functionality within a product or group of products. DSMs use rows and columns of components to convey component connectivity within a product or group of products.

During development of the repository, Bohm, et al. [2003] discovered that traditionally formulated functional models failed to completely capture each artifact’s complete functionality. By mapping the functional model of a product onto its components, it became evident that although some of the components did not have any conceptual level functionality associated with them, they were still essential to the overall functionality of the product. To represent this previously unaccounted functionality, supporting functions were formulated. These supporting functions convey information directly relevant to the structural interactions between components.

2-4. Automated Concept Generation

One method for automatically generating conceptual variants for a product design that is currently under development utilizes the Functional Basis to link component functionality with component compatibility and create, filter, and rank concept variants. [Bryant, et al, 2005]. Generated from a web-based repository of design information, the function-component matrix (FCM) and the design structure matrix (DSM) describe the function-component relationships and the component-component compatibility of existing consumer products. Product descriptions stored in the database allow access to information such as historical occurrence and failure mode, which help limit and rank design solutions. This design knowledge can be used to generate, filter, and rank concept variants. Functions comprising a proposed product’s functional model are mapped to lists of components that are capable of solving each function. The tree of possible component chains is then pruned by eliminating infeasible component connections according to component-component compatibility. The effectiveness of this pruning method relies heavily on having accurate component-component connections represented in the design repository. Figure 1 (see next page) briefly illustrates the philosophy behind the design tool and correlates the Functional Basis matrices used to generate and filter the conceptual design variants to the theory behind the concept generation algorithm. Additionally, having detailed knowledge about the structural interactions between artifacts will help augment the further development of similar automated techniques that would be able to include the indirect functionality of a product as well as the conceptual functionality of a product to produce a set of more complete conceptual design variants.

We hypothesize that a limited subset of functions from the existing Functional Basis can accurately represent all of the structural interactions within a product. Additionally, to capture both the flow interactions and structural interactions (or supporting functionality) of all the artifacts in an existing product, a new representation scheme is needed. In this paper, we present a new representation scheme able to assist designers in capturing the entire artifact relationships within a product. This representation, from here on referred to as an “assembly model”, borrows the flow representation technique from traditional functional modeling methods and incorporates the supporting functional information needed for concept generation. The assembly model graphically depicts all connections that exist between product artifacts and classifies each connection using functional terms defined in the Functional Basis. The assembly model bridges the gap between the conceptual functional model and actual functionality of an existing product. The current repository effectively supports information archival, storage, and reuse. The research presented here illustrates a graphical representation technique for representing product assemblies that facilitates data entry into the design repository and complements the existing design tools, making the repository a more powerful conceptual design tool resource.

3. Generating an Assembly Model

This work explores a method to accurately capture and represent both the flow interactions and the structural interactions of large component assemblies in order to support the conceptual
**Theory**

1. Assume we have the following chain of functions:

   ![Diagram of a chain of functions](image)

2. Assume the following components have the listed functionality:

<table>
<thead>
<tr>
<th>Component</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>f2, f3</td>
</tr>
<tr>
<td>C2</td>
<td>f1, f4</td>
</tr>
<tr>
<td>C3</td>
<td>f2</td>
</tr>
<tr>
<td>C4</td>
<td>f4</td>
</tr>
</tbody>
</table>

3. So, the chains of components that "solve" the functional model are:

   ![Diagram of component chains](image)

4. Assume the components have the listed compatibility:

<table>
<thead>
<tr>
<th>Component</th>
<th>Is Compatible With</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>C1, C2, C3</td>
</tr>
<tr>
<td>C2</td>
<td>C1, C4</td>
</tr>
<tr>
<td>C3</td>
<td>C1, C3, C4</td>
</tr>
<tr>
<td>C4</td>
<td>C2, C3</td>
</tr>
</tbody>
</table>

5. Limiting the possible component chains by compatibility, we get one plausible solution for our function chain:

   - **Viable Solutions**
     - C2 → C1 → C1 → C2
     - C2 → C1 → C1 → C4
     - C2 → C1 → C1 → C2

**Matrix Equivalent**

- **Connectivity Matrix**

- **Function-Component Matrix (FCM)**

- **Design Structure Matrix (DSM)**

*Figure 1*: Schematic of an automated method of concept generation. Matrix multiplication of vectors from the FCM leads to a series of matrices describing possible component design solutions (Step 3). Matrices created from multiplied rows of the FCM are then inserted into the Connectivity Matrix generated from the functional model. Cells of the possible design solution matrices are multiplied with corresponding cells of the DSM to produce a filtered list of design solutions (Step 5).
design phase. The graphical models contain data that is easy to visually extract and is useful for building computational resources for various concept generation techniques. A set of thirty-three consumer products with diverse functionality, varying part count, and differing energy domains were reverse engineered in order to record the structural and flow interactions between components. Using the component connection information, a modeling technique to graphically record the component layout and artifact interactions was developed. The recorded artifact connection data was then used to conduct a data analysis on the results. Finally, the assembly models were further analyzed to develop a design tool to help facilitate automated concept generation.

3-1. Assembly Model Formulation

An in depth investigation into thirty-three consumer products revealed that the supporting functionality (structural interactions) of the entire set of artifacts could be represented by a subset of four functions taken from the secondary level of the Functional Basis. These four functions, Couple, Secure, Position, and Guide, can either be used separately or in combination to completely describe the artifact connections contained within the product set. Using this information, a graphical model was devised to represent three important sources of information about the product’s assembly: 1) Component name and general orientation, 2) Flow interaction domain between artifacts, and 3) Structural interaction (i.e. supporting functionality) between artifacts. Labeled boxes are used to represent individual components and are roughly arranged to show relative orientation within the product assembly being modeled. Artifact boxes are then connected using lines of varying style that represent the three primary flow types defined by the Functional Basis [Hirtz, et al. 2002]. Line styles used to represent flow connections are synonymous with those used in the Functional Modeling technique [Pahl and Beitz, 1988]. For instance, energy flow connections are designated by a single weight line connecting two artifact boxes. Similarly, material flows connections are designated using bold lines, and signal flows connections are designated using dashed lines. The structural interactions, or supporting functionality, of each artifact is graphically labeled using four symbols representing one of each of the four Functional Basis functions found to describe the artifact interactions. Structural interactions are rigorously defined and exemplified in Section 3-2. An assembly model for the Dazey vegetable peeler (Figure 2a) was constructed using the steps outlined below, and is shown in Figure 2b:

1. Disassemble the product one assembly/component at a time, taking care to identify artifacts that directly interact with one another.
2. Represent each component as a box with an appropriate name. Boxes can be roughly arranged to represent relative artifact position within the product.
3. Denote the flow interactions that each artifact has with other artifacts as lines, using single weighted lines for energy domain connections, bold lines for material domain connections, and dashed lines for signal domain connections.
4. Using the definitions outlined in Section 3-2, identify structural connections (supporting functionality) using the four supporting functions by the following symbols: ◇ for a coupling connection, ● for a securing connection, ■ for a positioning connection, and ▲ for a guiding connection.

3-2. Rules for Selecting an Artifact’s Structural Connectivity

In order to create a repeatable methodology for assembly model generation, a rigorous set of definitions for structural con-
Connectivity must be established. The following guidelines should be followed when choosing labels for structural connection between artifacts to ensure accuracy and repeatability.

**Couple:** Two artifacts connected to each other with the aid of an intermediate artifact have a coupling connection. For example, in Figure 3(a), the rear housing and front housing are coupled together by a screw. The equivalent assembly model structure representation is shown in Figure 3(b). An artifact may couple or be coupled by more than one other artifact.

**Secure:** Two artifacts connected to each other without the aid of intermediate artifact and by an interference fit have a securing connection. For example, in Figure 4(a), the wire gauze secures the kernel container. The equivalent assembly model structure is shown in Figure 4(b). An artifact may secure or be secured by more than one other artifact.

**Position:** Two artifacts with a connection that restricts the movement of the positioned artifact in one or more directions but still allows the positioned artifact to come loose from the positioning artifact with little force have a positioning connection. For example, in Figure 5(a), the left housing positions the saw guide. The equivalent assembly model structure is shown in Figure 5(b). An artifact may position or be positioned by more than one other artifact.

**Guide:** Two artifacts whose mating surface involves a moving interface have a guiding connection. For example, in Figure 6(a), gear 2 guides gear 1. The equivalent assembly model structure is shown in Figure 6(b). An artifact may guide or be guided by more than one other artifact.

**Couple and Position:** Artifacts may be connected to each other by both a coupling and positioning connection. For example, in Figure 7(a), the bottom cover positions the middle case and both are then coupled together by an intermediate artifact (the screw). The equivalent assembly model structure is shown in Figure 7(b).

### 3-3. Consideration of Operation Order when Defining Structural Interactions

When examining artifact interaction, considerations must be made to three phases of operation within a product: the preparation, execution, and conclusion phases. The preparation phase consists of any product interactions that must be performed prior to the execution of the intended functionality of the product. The execution phase is in effect when the product is in use, performing its intended function. The conclusion phase consists of any product processes that are performed after the execution phase is complete. For some products, the type of connections between artifacts may change between the three phases of operation. When building an assembly model, connection type may be associated with the relevant operation phase through employment of a simple coloring scheme. An example of two artifacts whose interactions vary during the different phases of operation is shown in Figure 8(a). While the Presto Salad Shooter is chopping a vegetable (the...
execution phase), the blade guide maintains a secure connection to the hopper. After the execution phase is complete, the blade guide–hopper interaction becomes a guiding connection as the blade guide is separated from the hopper for blade removal and cleaning. The equivalent assembly model structure can be seen in Figure 8(b).

3-4. Application of Assembly Model Information to Support Concept Generation

The assembly modeling method presented provides good insight into the complete interactivity of components in existing products. Next, we will focus on translating this information into a form that supports integration into automated concept generation algorithms. Using the information gathered during the product exploration, a modification to the Design Structure Matrix (DSM) generated by the design repository was formulated. In addition to showing which artifacts are connected as the DSM does, this matrix, called the component connection matrix (CCM), shows all possible connection types between two components. A partial component CCM for the set of thirty-three consumer products is shown in Table 1.

The number 1 is entered in the corresponding column if the two components go together by that supporting function, otherwise, a zero is entered. In the CCM shown, the frequency of occurrence of a particular connection in the product set analyzed is designated by the number present in each of the cells. As in the standard DSM, the CCM results in a symmetric matrix. Contrary to the standard DSM form, though, the CCM shows how two artifacts might structurally interact with each other. In addition to filtering concept generation results, this information could be useful for planning designs that are in compliance with customary design for manufacture and assembly (DFM/DFA) techniques. For example, if the CCM identifies that some component A is compatible with another component B via either a coupling, securing, or positioning connection, the designer may opt to select the securing or positioning connection for the conceptual design. This design decision would comply with standard DFM/DFA techniques, as the coupling connection requires the addition of a third component. The CCM is a novel twist on an existing design tool that can be incorporated into computer code to help generate solutions for a complete product assembly.

4. Conclusions and Future Work

This paper outlines a novel function-based assembly model approach that captures a product’s component configuration and structural and flow interactions. Presented is a uniform representation of a product’s assembly that promotes the sharing of design knowledge without ambiguity. The component connection matrix (CCM), which is a modification of the existing design structure matrix (DSM) design tool, is a new addition in the area of product design able to assist the generation of feasible concept variants. Incorporation of the component connection matrix as a design tool could facilitate automated concept generation techniques that focus on generating complete assembly design solutions. From the CCM, feasible design alternatives that are in accordance with design for manufacture and assembly (DFM/DFA) practices can be readily identified. The assembly model methodology is effective.
Table 1: A partial component connection matrix (CCM) for the set of consumer products investigated during the course of this research.

<table>
<thead>
<tr>
<th>Comp/Comp</th>
<th>Container</th>
<th>Housing</th>
<th>Cover</th>
<th>Support</th>
<th>Electric Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container</td>
<td>3 0</td>
<td>1 0</td>
<td>5 1</td>
<td>3 1</td>
<td>0 0</td>
</tr>
<tr>
<td>Housing</td>
<td>1 0</td>
<td>21 9</td>
<td>12 3</td>
<td>19 10</td>
<td>9 11</td>
</tr>
<tr>
<td>Cover</td>
<td>5 1</td>
<td>12 3</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
</tr>
<tr>
<td>Support</td>
<td>3 1</td>
<td>19 10</td>
<td>1 1</td>
<td>9 3</td>
<td>2 1</td>
</tr>
<tr>
<td>Electric Motor</td>
<td>0 0</td>
<td>9 11</td>
<td>0 0</td>
<td>2 1</td>
<td>0 0</td>
</tr>
</tbody>
</table>

**Legend:**
- **Couple (C)**
- **Secure (S)**
- **Position (P)**
- **Guide (G)**

in both the development of new products as well as the redevelopment of existing products. Having knowledge about the flow and structure interactions between components at a designer’s disposal could effectively reduce the design cycle time and result in the faster launching of new products to market. Future work includes extending the CCM database to contain design knowledge for a large number of existing consumer products. Additionally, numerical values within the cells of the CCM can be expanded beyond the frequency of occurrence to incorporate various other design measures relevant to concept variant ranking (e.g. recyclability, manufacturability, failure, etc.) This numerical ranking would prove useful during the design evaluation stages of conceptual design. Further development could include the establishment of an automated method for automated translation of the graphical assembly model information for repository storage and reuse in the CCMs.

5. References


Bryant, C., McAdams, D., Stone, R., Campbell, M., Kuroglu, T., 2005, “Concept Generation from the Functional Basis of Design”, Accepted at ICED’05, Melbourne, Australia.


