AN ABSTRACT OF THE THESIS OF

Katherine N. Edmonds for the degree of Master of Science in Mechanical Engineering presented on March 20, 2020.

Title: A Preliminary Methodology For Automating Functional Modeling

Abstract approved: _____

Robert B. Stone Bryony DuPont

This thesis is the combination of two research publications working towards automating functional modeling. Functional modeling is an underutilized yet critical tool for concept generation and product design. Understanding the difficulty both novice and expert designers have in implementing functional modeling in their design process, this research sets out to streamline the process of functional decomposition and help designers include functionality in their designs. Using existing consumer product data from a Design Repository database, we developed an algorithm to find correlations between component and function and flow, returning component-function-flow (CFF) combinations. The automation process organizes these connections by component-function-flow frequency (CFF frequency), thus allowing the creation of linear functional chains.

The first publication explores a preliminary method to automate the generation

of linear functional chains using an Automated Frequency Calculation and Thresholding (AFCT) Algorithm. We use datasets of various scale and specificity to find correlations between functions and flows for components of products in the Design Repository. We use the results to predict the most likely functions and flows for a component, and then verify the accuracy of our algorithm by cross-validating a subsection of the data against the automation results. We then apply existing grammar rules to order the functions and flows in a linear functional chain.

The second publication describes the methodology used to develop a new metric, which we refer to as weighted confidence, to provide insight on the fidelity of the data returned by the above AFCT algorithm. In the previous publication, we found that CFF frequency is the best metric in formulating the linear functional chain for an individual component; however, we found that this metric did not account for prevalence and consistency in the Design Repository data. The weighted confidence metric is calculated by taking the harmonic mean of two metrics we extracted from our data, prevalence, and consistency.

Improving these automation results, allows us to further our ultimate objective of this research, which is to enable designers to automatically generate functional models for a product given constituent components.

A Preliminary Methodology For Automating Functional Modeling

by

Katherine N. Edmonds

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 20, 2020
Commencement June 2020

Master of Science thesis of Katherine N. Edmonds presented on March 20, 2020.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

---

Dean of the Graduate School

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

Katherine N. Edmonds, Author

PUBLICATION THESIS OPTION

This thesis is presented in accordance with the Manuscript Document Format option. Two manuscripts are provided. The first was submitted for publication to the Design Society Ninth International Conference on Design Computing and Cognition (DCC) 2020, and the second was submitted for publication in the ASME 2020 International Design Engineering Technical Conferences (IDETC/CIE 2020).

ACKNOWLEDGEMENTS

I want to express my gratitude and appreciation to my advisor Dr. Robert B. Stone. He has been extremely supportive and has helped me find multiple opportunities to expand my knowledge and expertise.

I would also like to thank my co-advisor Dr. Bryony DuPont. Dr. DuPont kindly accepted me into her lab group and made me feel welcome. She challenged me, but also had great confidence in me, when I did not. I would also like to thank Dr. Chris Hoyle for his input and knowledge on this project as a valued committee member. I would like to thank all the professors and my colleagues at the Design Engineering Lab. Thanks to their help, support, and motivation, I was able to complete this project.

I want to give a special thanks to my wife Marcía Snyder for being so patient with all my scattered interests and for whole heartily supporting me in this latest endeavour. Last but not least, I would like to thank my sweet adorable parents, Larry and Sue Edmonds, who have helped give me lots of opportunities to learn and grow throughout my life.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDIX FIGURES

Chapter 1: Introduction

The concept generation phase of the design process is an area where the most creativity and innovation occur [47]. This early design phase is where the least cost for changes occurs, so there is significant research on tools and tactics to improve the concept generation phase and how to improve the efficiency of the process. There are multiple tools that product designers use during this phase to help guide the creative process towards a viable concept. One of these tools is deriving the functionality of the product through a functional decomposition, graphically represented by a functional model [45] [2]. Consistency and uniformity are built into the process with the widely accepted use of the Functional Basis and Component Basis terms [40] [17] [23]. However, even with consistency built into the process, functional models can vary widely by the individual user input [29]. Additionally, functional modeling if often overlooked or omitted from concept generation because designers have a difficult time considering design in terms of the functionality of the product. Rather designers are more comfortable with component-based solutions, often benchmarking existing products [28]. Incorporating functional modeling early into the design phase can help the shift of resources in the project lifecycle to earlier in the design process when the cost of making changes is low, but the impact of those changes is high.

1.1 Motivation

In order to encourage more widespread use of functional decomposition in the design process, we explore the fundamental underpinnings necessary for a data driven method for automating functional modeling. We utilize data from an existing design repository. Mining this data in Oregon State University's Design Repository for the existing correlations between component, function, and flow of constituent product components allows us to connect function to product component, information that can provide the designer with the functional breakdown of components. The motivation for automating functional modeling is three-fold, increasing the use and comprehension of functional modeling, improving the Design Repository by expanding the data and streamlining the process of adding products, and connecting components to function and flow to allow for the inclusion of function in component-based design.

1.2 Glossary of Terms

Throughout this thesis, there are terms used that may have alternative meanings or for conciseness are defined with an acronym. Therefore, Table 1.1 provides a list of definitions of terms and acronyms used in this thesis.

Table 1.1: Terminology Utilized Throughout This Thesis

Term	Definition
The Design Repository	A data-base containing 142 consumer products at different levels of abstraction including artifact, component, function and flow.
Product Function	Characterized in a verb-object (function-flow) format [40].
Function	A description of an operation to be performed by a device or artifact, expressed as the active verb of the sub-function [40].
Flow	A change in material, energy or signal with respect to time. Expressed as the object of the sub-function, a flow is the recipient of the functions operation [40].
Component	The lowest level of a system or product in the repository.
Component-Function-Flow (CFF)	The correlated function and flow with each individual component found in the Design Repository.
Data-Driven Design	Methodologies for extracting information and insights from data and existing research to improve design processes.
Linear Functional Chain	A component-based linear section of a full functional model.
CFF Frequency	The measure of how often a CFF combination appears in a given dataset.
Automated Frequency Calculation and Thresholding (AFCT) Algorithm	An algorithm that returns the CFF frequency of the component-function-flow (CFF) combinations and applies a threshold the returned only the top 70% of functions and flows per component.
Prevalence	The measure of the commonness of the component.
Consistency	The measure of how uniform the CFF combinations are per component.
Weighted Confidence	The harmonic mean of the two metrics, prevalence and consistency, which is a measure of the AFCT data fidelity

1.3 Contributions

This thesis consists of two manuscripts that describe the beginning stages of automating functional modeling. The presented methodology uses an Automated Frequency Calculation and Thresholding (AFCT) algorithm to find the component-function-flow (CFF) frequency of the CFF combinations in a Design Repository. The results of this algorithm are improved by using a weighted confidence metric, which was developed based on the consistency and prevalence of the CFF frequency data. The contributions to the field are as follows:

1. *The development of an algorithm to automatically create linear functional chains based on the most likely functions and flows per component.*

   The AFCT algorithm orders the CFF frequencies from largest to smallest, sums the frequencies of each CFF combination, and uses a threshold of 70% of the sum of frequencies of combinations for each component. These likely functions and flows can be turned into linear function chains using existing grammar rules and designer expertise. This algorithm can be applied to the data in the Design Repository by designers who are interested in the connection between component and function and flow.

2. *The development of a weighted confidence metric to improve data fidelity of the AFCT algorithm.*

   The contribution of this metric is not to replace CFF frequency as a method of finding the most likely component-function-flow correlations but to improve the reliability of the automation results by providing additional infor-

mation about the data fidelity from the weighted confidence metric.

3. *The formation of methodology for weighting any dataset with a wide range of individual occurrences.*

   The weighted confidence metric can be adopted and applied to any dataset. The framework of this metric allows researchers to include rare data or other outlying data by indicating the prevalence and consistency of the data being utilized.

4. *Expanding and improving the data in the Design Repository.*

   This research is the beginning steps of fully automating functional models, which will allow new products to be more easily added to the repository by a variety of users without requiring functional modeling expertise and minimizing the concern of consistency in data entry. The streamlining of the process of adding new products with automating functional modeling allows not only individual products to be added by users but also the addition of entire repositories.

Chapter 2: Background

2.1 Functional Decomposition

A functional model is the graphical representation of the functional decomposition of a product, and example of a Black and Decker Dustbuster can be seen Figure A.1 in the Appendix, which demonstrates the complexity of functional models. Novice and experienced designers all have difficulty building functional models because of the complicated process, and this can lead to the functional modeling process being entirely omitted from the concept generation phase. Yet we know that concept generation is more complete when function is considered [45]. While there are significant research and information available on how to build functional models [45] [2], Sridharan and Campbell point out that even though there is a formal language, the Functional Basis [40] [17], repeatability is still difficult among students and researchers[20]. In order to help solve the consistency issue with building functional models, Nagel et al. developed grammar rules. Additional researchers, Sridharan and Campbell and Bohm and Stone, developed grammar rules and tested their rules with students building functional models. The students given the grammar rules created more consistent functional models and had a more full understanding of functional decomposition than the students without the grammar rules[38] [4]. Bohm and Stone developed rules associated with individual

functions and dictate the allowed incoming and outgoing flows [4].

To simplify the process of automation, we have chosen to begin by building individual component based linear functional chains rather than full complex functional models. We have shown in previous research that finding associations between functions and flows and component allows us to build these linear functional chains [44]. Building these linear functional chains, we found several of the grammar rules in the previous research applicable to our current research. We will apply the grammar rules to the component-function-flow combinations to determine the appropriate order of the functions and flows for the linear functional chains. Starting with this simplified model, we can work out the issues and problems with automation rather than starting with such complexity as a full product functional model.

2.2 Design Repositories

A design repository is a product database where data can be searched and retrieved at different levels of abstraction to help improve design knowledge and data-driven design decisions[43]. A well-populated repository with different levels of abstraction offers designers a wealth of information to aid in decision making. The Design Repository[1] we are using is comprised of 142 consumer-based electro-mechanical products and is housed online through the Design Engineering Lab at Oregon State University. The design information of each product can be divided into seven main

[1]The Design Repository is a database of design information. A basic web interface is available at ftest.mime.oregonstate.edu/repo/browse

categories: artifact, function, failure, physical, performance, sensory, and media-related information types. In its current form, the repository has a tools section, where designers can select single or multiple products from the list of products and create three different outputs from the information. The first output is the Design Structure Matrices (DSM), which is a matrix representation of the assembly model [8]. The second output is Function Component Matrices (FCM), which is a matrix representation of the function and component relationship [22]. The last output is the Configuration Flow Graph, which connects the flows to each other (CFG) [21]. These tools provide the user with a wealth of information. However, in its current form the repository can be difficult to navigate for novice users; additionally, users are not able to input new products.

Additional design repositories include bio-inspired repositories such as asknature.org, IDEA-INSPIRE, and DANE (Design by Analogy to Nature Engine)[15]. These tools are used by designers to inspire creative and innovative solutions based on function derived in biology. Product data management (PDM) systems are also similar databases to a design repository. PDMs store and retrieve the product and part data, revision history, bill of materials, and CAD models [42]. PDMs are typically proprietary software and used internally by industry, in contrast the Design Repository is publically accessible. While helpful for innovation and information, these repositories do not provide the user with the amount of detailed function based information that the OSU design repository stores. The database structure of the OSU repository also provides mapping and connections between the categories of the product systems, such as function and component, shown in Figure

5.1.

As we develop our automation process, it becomes easier in the future to add information from other repositories such as the ones listed above, which significantly expands our database. We are working with additional OSU researchers to house the information from an existing Sustainable Design Repository in the OSU Design Repository [14]. Combining this information adds additional products, as well as sustainable design information such as LCA analysis and manufacturing processes. Expanding on the work presented in the Function-Human Error Design Method (FHEDM), Soria et al. have been using Design Repository data to develop new relationships, such as incorporating the user, user interactions, human error [48] [37]. The database structure of the Design repository provides mapping and connections between categories of the product systems, expanding these connections to include sustainability and user-system interactions will bring these important considerations to the early phase of design decisions.

Chapter 3: Related Research

In addition to the two publications presented in this thesis, I assisted students Melissa Tensa and Alex Mikes with research related to automating functional modeling. Below briefly outlines two publications that resulted from this work. The first was authored by Melissa Tensa and submitted to ICED 2019 the 22nd International Conference on Engineering Design titled *Toward Automated Functional Modeling: An Association Rules Approach for Mining the Relationship between Product Components and Function*. The second was authored by Alex Mikes and submitted to IDETC/CIE 2020 The International Design Engineering Technical Conferences titled *Optimizing an Algorithm for Data Mining a Design Repository to Automate Functional Modeling*.

3.1 An Association Rules Approach for Mining the Relationship between Product Components and Functions

This thesis builds on previous work using the association rules to find correlations with repository data [44]. In this work, we found associations between component-function-flow on a smaller dataset found in the repository (12 Black and Decker products). We used the Apriori algorithm to find the associations between component and function-flow. Association rules describe the relationship between items

in item sets. The typical application of association rules is in a market-based analysis. For example, in a supermarket, 90% of customers who buy bread and butter also buy milk [1]. The Apriori algorithm is often used to find these types of associations. Well-documented, the algorithm is useful for dealing with large datasets and iteratively looks for frequent itemsets [30]. The Apriori algorithm outputs three measures; support, confidence, and lift. Thresholds of each measure set by the user control the output of the algorithm and can influence the results. Agrawal suggests the support threshold should be set high enough to ensure the relationship is significant [1].

Support determines the probability of the prevalence of an item within all of the itemsets [26]. Confidence determines the probability of two items appearing in the same itemset. Lift, known as the interestingness measure, is the ratio of the support of the association and the support values of the individual items within the itemset. While all three measures provide information, we were most interested in the confidence of the CFF associations. An example of how confidence can be described in our dataset is *what is the probability that the component battery associated with the function-flow store electricity compared to other function-flow combinations that appear for the component battery?*

For this work, we created a learning set of Black and Decker consumer products found in the Design Repository to find the CFF combinations using association rules. We compare the CFF combinations found in the learning set to a verification product case, the Delta jigsaw.

We first retrieved our selected data by querying the repository database. We

then used the Apriori algorithm to determine the probabilities of associations between components, functions, and flows. The Apriori algorithm output the three measures of association: *Support*, *Confidence*, and *Lift*. Table 3.1 shows a sample of components and the Apriori algorithm results. The table displays a x if the function and flow were also found in the verification product, the Delta jigsaw. This verification product helps determine the accuracy of the results from the algorithm; the verification product was not included in the original learning dataset. The results showed some similarities between the CFF combination associations of the Black and Decker dataset and the Delta jigsaw verification product. However, we learned that our results were limited due to the size of the data set and the choice to only use one product in our verification process.

Based on these findings, we improved upon this research method in our forthcoming work. First, we decided to expand both our learning and verification datasets to test the robustness of our methodology. Additionally, since confidence was the primary metric we found useful in our data analysis, we decided to rely only on the confidence data, which we refer to as CFF frequency in all subsequent research, moving away from using association rules. We also implemented a 70% threshold to return the most likely functions and flows rather than all of the results. I describe this expanded research in 4

Table 3.1: Sample of Verification Dataset [44]

Head	Body	Confidence (%)	Support (%)	Lift (%)	Jigsaw
Blade	export mechanical	16.67	0.35	1443.33	
	separate solid	16.67	0.35	1202.78	×
	export solid	16.67	0.35	759.65	×
	import solid	16.67	0.35	601.39	×
	transfer mechanical	16.67	0.35	313.77	×
	guide mechanical	5.56	0.12	481.11	
	secure solid	5.56	0.12	54.67	
	guide solid	5.56	0.12	45.82	
Cam	change mechanical	66.67	0.23	2510.14	×
	transfer mechanical	33.33	0.12	627.54	×
Electric Wire	transfer electrical	66.13	4.73	938.82	×
	secure solid	17.74	1.27	174.60	
	position solid	12.90	0.92	87.99	
	guide solid	3.23	0.23	26.61	
Screw	couple solid	93.85	7.04	913.15	×
	position solid	6.15	0.46	41.96	
Spring	position solid	47.37	1.04	323.00	
	guide solid	21.05	0.46	173.63	×
	store mechanical	10.53	0.23	4557.89	
	supply mechanical	10.53	0.23	4557.89	
	secure solid	5.26	0.12	51.79	
	couple solid	5.26	0.12	51.21	

3.2 Optimizing an Algorithm for Data Mining a Design Repository

The optimization process outlined in this publication is designed to improve our AFCT algorithm. Here we investigated ways to find the optimize the threshold values used in the algorithm. In the AFCT algorithm, we have been setting the classification threshold at 70% to based on the Pareto Frontier from the Form Follows Form method [4]. However, we know that this may not be the most optimized threshold. We use the AFCT algorithm to retrieve data for the optimization process. This methodology is described in depth in 4. To determine the optimized threshold, we iterated through the verification process using 18 different values of

classification thresholds (10, 15, 20, 25. . . 95%) [27]. The optimum value for the classification threshold was found to be 55%. This research was completed after both publications presented in this thesis, so the threshold used in the AFCT algorithm in this work remains 70%. The optimized 55% threshold result can be used in future work. Additional future work could include optimizing the classification threshold separately for individual components or sub-assemblies. Note: Additional optimization research was published in this paper, but is not directly related to the research in this thesis, and therefore is not discussed.

3.3 Summary of Manuscripts

The first manuscript presented in this thesis is the *Data mining a design repository to generate linear functional chains: a step toward automating functional modeling*, submitted to DCC 2020 and found in chapter 4. The objective of the first manuscript is to further explore a preliminary method to automate the generation of the linear functional chains of components using the AFCT algorithm.

The second manuscript presented in this thesis is the *A Weighted Confidence Metric to Improve Automated Functional Modeling*, presented at IDETC 2020 and found in chapter 5. The objective of this manuscript is describe a methodology for developing a weighted confidence metric to improve the data fidelity of the AFCT algorithm.

Chapter 4: Data mining a design repository to generate linear functional chains: a step toward automating functional modeling

Authors

Katherine Edmonds

Design Engineering Laboratory

School of Mechanical, Industrial &

Manufacturing Engineering

Oregon State University

Email: edmondka@oregonstate.edu

Alex Mikes

Design Engineering Laboratory

Email: mikesa@oregonstate.edu

Bryony DuPont

Design Engineering Laboratory

Email: Bryony.DuPont@oregonstate.edu

Robert B. Stone

Design Engineering Laboratory

Email: Rob.Stone@oregonstate.edu

Proceedings of the 2020 ASME International Design & Engineering Technical Conferences

42st Design Automation Conference (DAC)

IDETC 2020

August 16-19, 2020, St. Louis, MO, United States of America

Ninth International Conference on Design Computing and Cognition - DCC'20 29 June1 July 2020. Georgia Institute of Technology, Atlanta, USA. Preceded by Workshops. 27-28 June 2020.

4.1 Abstract

Populating the different types of data for a design repository is a difficult and time-consuming task. In this work, we report on techniques to automate the population of data related to product function. We explore a preliminary method to automate the generation of the functional chains of components from new products based on hierarchical data from an existing design repository. We use datasets of various scale and specificity to find correlations between functions and flows for components of products in the Design Repository. We use the results to predict the most likely functions and flows for a component, and then verify the accuracy of our algorithm by cross-validating a subsection of the data against the automation results. We apply existing grammar rules to order the functions and flows in a linear functional chain. Ultimately, these findings suggest methods for further automating the process of generating functional models.

4.2 Introduction

Product design in engineering is a well-studied process [31], yet many aspects remain difficult and hard to define after decades of research, especially the early

stages of concept generation. However, that concept generation phase is the one part of the design process where there is the most room for creativity and innovation [47]. Additionally, the concept generation phase is the least costly time of the design process to integrate major changes[45] and exploration during this phase should be encouraged. We use the term designers broadly to refer to those who are working in their field to develop new concepts or products, as well as iterating on existing concepts and products. During concept generation in product design, designers focus on gathering accurate customer needs, determining engineering specifics, deriving the functionality of the intended product, and ideating potential form solutions.

Functional decomposition is a well-known abstraction technique that allows designers to develop a graphical representation of a products functionality as a functional model [45] [2]. There has been extensive work done to develop consistency in the nomenclature, beginning with the development of the Functional Basis terms [40] [17]. However, as Nagel et al. [29] point out, there is still inconsistency in user input in the structure of functional models. Novice and experienced designers all have difficulty building functional models because of the complicated process, and this can lead to the functional modeling process being entirely omitted from the concept generation phase. Yet we know that concept generation is more complete when function is considered [45]. Figure 4.1 shows how incorporating functional modeling early into the design phase can shift the majority of resources in the project lifecycle earlier in the design process when the cost of making changes is low, but the impact of those changes is high. While the above is true, designers are

often more comfortable with component-based solutions, and tend to focus more on components rather than the functionality of a sub-assembly or product. This type of design often benchmarks existing products during the concept generation phase [28].

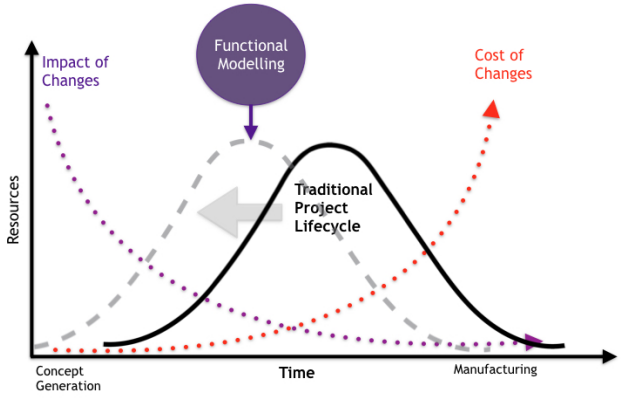


Figure 4.1: Cost Analysis Over Time During Concept Generation

Eckert and Stacey [11] designate the term source of inspiration for the conscious use of previous designs in the design process. Design repositories can provide designers with data at multiple levels of abstraction, such as components, functional representations (e.g. functions and flows), or high level customer needs responses, offering a central location for source of inspiration products. Our research uses data from an existing design repository, know as the Design Repository, to support this type of reuse in design. Our data-driven design (DDD) approach leverages research from the multi-decade long project developing a design repository [43] [6] [8] [22] [38]. For our purposes, we define data-driven design as methodologies for extracting information and insights from data and existing research to improve

design processes [34].

We utilize the extensive previous work on the repository and expand the most recent work; the Form Follows Form approach [4] [5]. The Form Follows Form (FFF) approach is based on the concept that most designers think in terms of components rather than function when in the concept generation phase. Utilizing this concept, we capture the underlying functionality of the chosen components using data from the design repository [5]. In the FFF approach, Bohm et al. calculated the frequency of function and flow associated with components *separately*. Our research continues to build on this concept by developing a *combined* association between component and function-flow of CFF, to predict the most likely functions and flows associated with each component. We choose to only consider the incoming flows to simplify our analysis, as we found in analyzing our datasets that less than 5% of the results have different inflow and outflow. With this data, we build linear functional chains for components. These linear functional chains will ultimately help us develop a method for automating the creation of functional models. As this research develops, machine learning from the *combined* CFF combinations is anticipated to help eliminate errors such as illogical or impossible CFF combinations in attempting to combine them later during functional modeling automation.

There is significant research on developing consistency in the grammar and syntax of functional models [40] [17][22]. The Design Repository has this consistency in language built into the data, for example, functions are entered using the Functional Basis terms [40], and components with the Component Basis terms [3]. This

terminology allows us to create correlations that remain consistent throughout the datasets.

Our immediate research objectives are to 1) mine the design repository for datasets 2) calculate frequencies of CFF combinations and apply a classification threshold with an automation algorithm, 3) validate the accuracy of the automation algorithm, and 4) apply existing rules to develop linear functional chains based on our findings.

4.3 Background

4.3.1 The Design Repository

A design repository is a product database where data can be searched and retrieved at different levels of abstraction to help improve design knowledge and data-driven design decisions[43]. A well-populated repository offers designers a wealth of information to aid in decision making. The Design Repository[1] we are using is comprised of 142 consumer-based electro-mechanical products and is housed online through the Design Engineering Lab at Oregon State University. Each product is divided into seven main categories of design information: artifact, function, failure, physical, performance, sensory, and media-related information types. A visual reference of the data schema (i.e., the connections between data) is shown in Figure 5.1 [6].

[1]The Design Repository is a database of design information. A basic web interface is available at ftest.mime.oregonstate.edu/repo/browse

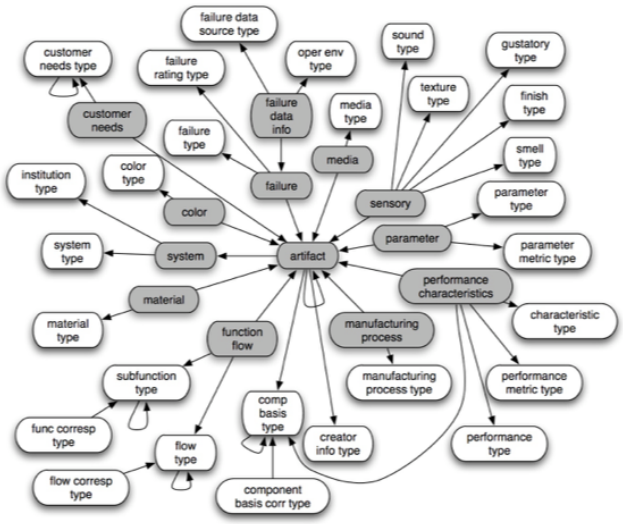


Figure 4.2: Design Repository Data Schema [6]

While there is significant research and information available on how to build functional models [45] [2], Sridharan and Campbell point out that even though there is a formal language, the Functional Basis [40] [17], repeatability is still challenging among students and researchers[20]. To help solve the consistency issue with building functional models, Nagel et al. [29] developed grammar rules. When applied correctly, these grammar rules help determine the appropriate order of the linear functional chains. Additional researchers, Sridharan and Campbell and Bohm and Stone, developed grammar rules and tested their rules with students building functional models. The students, given the grammar rules, created more consistent functional models, and had a better understanding of functional decomposition than the students without the grammar rules[38] [4]. Bohm and Stone

developed rules associated with individual functions and dictate the allowed incoming and outgoing flows [4]. We found several of the grammar rules in the previous research applicable to our current research. We apply these grammar rules to the data returned from the automation algorithm.

4.3.2 Machine Learning and Data Mining

Data mining and machine learning are general terms that refer to many different techniques of using information to predict results. The work that we are doing is considered data mining because our algorithms extract knowledge from the data and do not alter it based on the findings, which would be considered machine learning. However, we borrow some of the terms and methods that are traditionally applied to machine learning problems to find patterns within our data.

A classifier is an algorithm learns from data, finds patterns within it, and then predicts whether something is or is not within a class. An example is a classifier that predicts whether or not an email is spam[9]. The machine learning algorithm looks at examples of emails that a person has labeled as *spam* or *not spam* and finds patterns within them to label any other email as *spam* or *not spam*[24]. The accuracy of this classifier is quantified by testing it against other emails labeled as *spam* or *not spam* and recording which predictions were correct and incorrect.

We are using data mining techniques to find the frequency of occurrence of CFF combinations in products in the design repository. We use that frequency information to predict what functions and flows a component will have in a new

product.

4.3.2.1 Frequency

In our previous work [44], we used the Apriori algorithm to find associations between component and function-flow. During data analysis, we found that using association rules was excessive for the results we wanted to obtain. We simplified our calculations, focusing on the frequency of CFF combinations, which is numerically equivalent to the confidence metric from association rules. The Form Follows Form approach uses a similar method of calculating the frequency of the function and flows correlated with each component [5].

Frequency determines the probability of two items appearing in the same itemset. In our datasets, we find the frequency by calculating how often a component and function and flow appeared together. The frequency values for all CFF combinations for each component sum to 100%, regardless of the number of functions and flows per component. For example, the CFF combination *screw* and *couple solid* appears in the consumer products dataset 589 times out of 647 total CFF combinations for the compenent screw, so the frequency of that combination is 589/647 or 91%.

Sometimes a CFF combination may only appear once if it is an unlikely combination or is a specialized component or function only appearing in one product in the repository, such as *pressure gauge* and *indicate mechanical*. In these cases, the frequency that the CFF combination occurs is 100%.

4.3.2.2 Threshold

In our work, we are predicting the functions and flows for components, and our threshold is a cutoff that predicts that the top 70% of functions and flows would be likely for future components. Our automation algorithm orders the frequencies from largest to smallest, sums the frequencies of each CFF combination, and applies a 70% threshold to each component. This algorithm is different from a traditional classifier that would discretely label a class based on individual probability. This 70% threshold was developed based on previous research by Bohm, who found that 70% of functions and flows are realized within the first 30% of unique instances of a particular component, which he credited to the Pareto optimal gaming theory [4]. We found in our data analysis that the 70% threshold is often the point where adding additional functions and flows for a component contributed a negligible delta to the sum of frequencies and decreased the accuracy of the automation results.

4.3.2.3 Cross-Validation

A common method to find the accuracy in a machine learning classifier is known as cross-validation, which withholds a subset of data from the initial set, so the machine learning algorithm does not learn from this subset. This subset is then used to find the accuracy of how well the classifier performed at predicting results [16]. Testing with data from which the classifier did not learn is essential for reducing bias in the results [39]. The subset of withheld data is known as the

testing set and the rest of the data that the machine learning algorithm processes is known as the *training set*.

Due to the variability in data, cross-validation is often performed multiple times with different testing and training sets and averaged over all iterations. Kohavi found that 10-fold cross-validation produces the best results for most applications even when additional computational power is available [19], so we use this method to determine the accuracy of our automation algorithm using the metrics of precision, recall, and the F1 Score. The general method is known as *k-fold cross validation*[10].

In our previous work, we used a single product (a Delta jigsaw) for our *testing set*, and the Black and Decker dataset was our *training set* [44]. This initial exploration helped us gain valuable insight into the initial stages of this process, but cross-validation is a more robust method.

4.3.2.4 Precision, Recall, and the F1 Score

The method and effectiveness of calculating accuracy varies based on the type of data being used. Simple accuracy is calculated as a ratio of correct responses to total responses. In our case, a correct response is when the data mining algorithm finds a function-flow combination for a component that matches the *testing set*. Simply counting correct responses misses some of the additional ways in which the automation can be wrong. Precision, recall, and the F1 score account for these cases by using the confusion matrix shown in Table 4.1 to calculate ratios of the true

positives, false positives, and false negatives [36]. Note that true negatives are not included in the calculation for metrics because, for many systems, including ours, most results are true negatives, and including these in our accuracy calculations would highly increase our results and make the classifier appear to be performing better than it is.

Table 4.1: Accuracy Confusion Matrix

		Predicted CFF?	
		Yes	No
Actual CFF?	Yes	True Positive	False Negative
	No	False Positive	True Negative

Precision is the ratio of correct CFF combinations to all CFF combinations identified by the automation algorithm (Equation 4.1). This number is the ratio of CFF combinations that were identified as being in the product that are actually in the product.

Recall is the ratio of correct CFF combinations to all CFF combinations found in the automation algorithm (Equation 4.2). This number is the ratio of the actual CFF combinations that were correctly predicted.

The F1 Score is the harmonic mean of precision and recall that equally balances the importance of the two metrics and punishes extremes (Equation 4.3). F1 is a more powerful metric than simple accuracy and provides a better analysis of the ability of an automation algorithm to predict results.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{4.3}$$

4.4 Methods

In this work, we mine the design repository for data to find the most likely functions and flows correlated with each component for several datasets. We refer to this correlation as component-function-flow (CFF). We are building on previous work using association rules, where we found associations between component-function-flow on a single dataset (12 Black and Decker products) [44]. Here, we expand our learning datasets as well as our validation methods. We chose three data subsets from the repository driven by component: 23 products with the component **heating element**, 32 products with the component **blade**, and 44 products with the components **container/reservoir**. We applied our automation algorithm (described later) on each dataset separately and calculated the accuracy of its ability to predict function and flows for an input component. We then compared the accuracy results of each of the three data subsets to a dataset containing all 142 **consumer products** from the design repository and an additional subset containing 12 products that were all made by **Black and Decker**.

We chose products with the **heating element, blade**, and **container/reservoir** components in an attempt to single out products with similar functionality. We chose the Black and Decker products because this was the most extensive dataset

available to provide a company product portfolio, which offers a subset of products based on construction rather than functionality. *We hypothesize that narrowing the dataset to functionality based on component will yield more accurate function and flow prediction results.* The Black and Decker and consumer product datasets serve as reference datasets to test our theory. We developed an automation algorithm in Python, which calculates and sums the frequency of each CFF appearing in each dataset. The algorithm applies a classification threshold to the top 70% sum of frequencies for each component. The correlations found within the 70% threshold in our data mining process can then be used to predict the linear functional chain of a component.

Step 1. Retrieve Datasets. To extract information, we query the repository to create five test datasets: 1) all **consumer products**; 2) all **Black and Decker** consumer products to represent a general product family by the same manufacturer; and three subsets of consumer products with 3) **heating element**, 4) **blade** and 5) **container/reservoir** as a component in the assembly to represent products with a similar component and functionality. We chose to combine reservoir and container into one dataset because of the similarity of functionality, and combining them allowed us to have a similar size dataset as the other two component-based datasets.

Step 2. Automated frequency and thresholding Next, we apply an automation algorithm to each of the five datasets, implemented in Python v3.7. First, the algorithm calculates the frequency of the functions and flows for each component in the input dataset; then, those values are sorted from largest to

smallest, summing to 100%. The threshold is applied to capture the top 70% of the sum of frequency values for each component in each dataset, based on the Pareto Frontier from the Form Follows Form method [4]. The results from the electric cord component from the blade dataset provide a simple example in Table 4.2.

Table 4.2: Example dataset to illustrate threshold automation for the component Electric Cord

Electric cord	**Frequency delta**	**Running sum of frequency**	**Threshold**
Import electrical	0.35	0.35	Keep
Transfer electrical	0.3	0.65	Keep
Export electrical	0.15	0.8	Keep
Position solid	0.12	0.92	Reject
Couple solid	0.08	1	Reject

For the electric cord example, the frequency of the first two functions and flows sums to 65%, so the third is added to the list to reach the 70% threshold, which brings the sum to 80%. Our method assumes that capturing approximately 70% of the total frequencies will begin to give an accurate representation of the functions and flows that a component usually performs. Additionally, this Pareto optimal threshold is the point where adding additional functions and flows for a component usually contributed a negligible delta to the sum of frequencies and increased the error in the automation results.

Step 3. Cross-Validation. As a means of verifying the accuracy of the automation algorithm, we use a 10-fold cross-validation method to find the precision, recall, and F1 score of each iteration. The design repository categorizes products

by an identification number, which we randomize and separate into ten folds.

For example, the blade product dataset contains 32 products. This number is not divisible by ten without a remainder, and our data requires each product to remain intact, so we actually have eleven folds. Ten folds have three products each, and the eleventh fold has the remaining two products.

We apply the frequency calculation and thresholding algorithm three times for each dataset that we queried. The first validation is a traditional cross-validation and finds the accuracy of the automation algorithm when the training set comes from the component-specific dataset (blade, heating element, reservoir/container), and the testing set is also the component-specific dataset. The second validation finds the accuracy when the training set is the consumer products dataset, and the testing set is the component-specific dataset. The third validation uses the Black and Decker dataset as the training set and the component-specific dataset as the testing set.

We stray from traditional cross-validation in two of three of these validation tests by selecting the folds for the testing set and training set from different datasets. This method gives us a cross-reference for accuracy between datasets and allows us to see if one dataset is better at predicting results for itself or for another dataset. The three variations of accuracy testing are shown in Table 4.3. These three validations were performed for each of the three component-specific datasets, resulting in nine F1 scores.

One of the benefits of traditional cross-validation is that the testing set is withheld from the training set to reduce bias in the results. With this method of

validation, when the testing set and training set come from different datasets, the folds contain some overlapping data. To combat bias, we made sure to remove all products in the fold for the testing set that were also in the folds for the training set.

Table 4.3: Validation Cases

Validation Number	**Testing Set Source Dataset**	**Training Set Source Dataset**
1	Component-specific	Component-specific
2	Component-specific	Consumer products
3	Component-specific	Black and Decker products

Step 4. Apply Grammar Rules to Determine Linear Functional Chain. After analyzing and organizing the results of the top 70% of the functions and flows, we apply the grammar rules described in section 2.1 to the results to determine the linear order of the functions in the functional representation. For the electric cord example in Table 4.2, the three functions are import, transfer, and export. They all have the same flow of electrical energy between them. The grammar rules developed by both Nagel et al. [29] and Bohm et al. [4] state that the import function occurs first and only once per flow in a chain of components, so it is placed first. The grammar rules also state that export is the last function in a chain of components, which leaves transfer as the middle function in this chain. A visualization of this process can be seen in Figure 4.3.

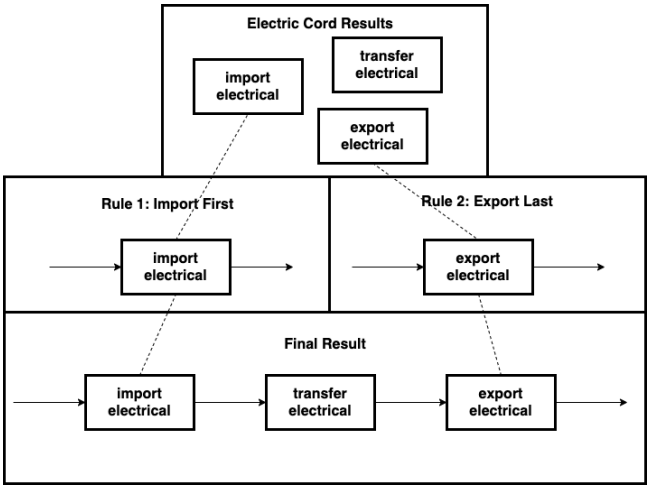


Figure 4.3: Example of Grammar Rule application

4.5 Results

4.5.1 SQL Query

The results of our SQL query can be seen in Figure 4.4. The total number of CFF combinations is the number of times a component has a particular function and flow regardless of the number of times they repeat in the dataset. The number of unique combinations is the number of times a component has a particular function and flow at least once, and additional instances of that combination are no longer unique. The number of products in each dataset can be seen in Table 4.4.

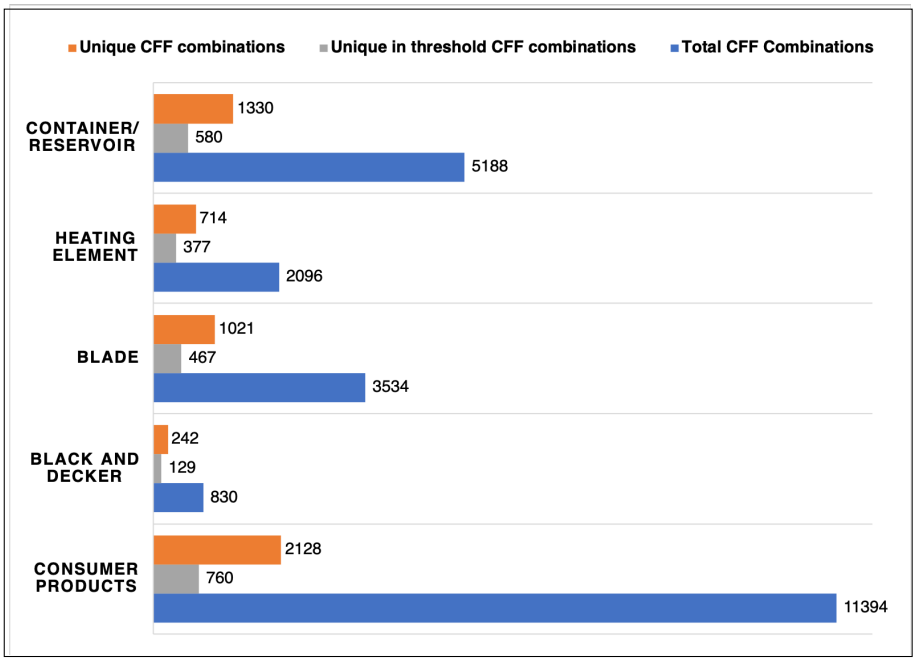


Figure 4.4: Query Results

4.5.2 Automated Frequency Calculation and Thresholding Algorithm

The algorithm returned CFF combinations for all five datasets filtering out the combinations that were above the threshold. Figure 4.5 shows the CFF combinations for four components in the consumer products dataset, results above the black line are the CFF combinations found within the threshold. As seen in Figure 4.5, the 70% threshold is often the point where adding additional functions and flows for a component contributed a negligible delta to the sum of confidence. In order to remain consistent, all of the results are taken from the consumer database.

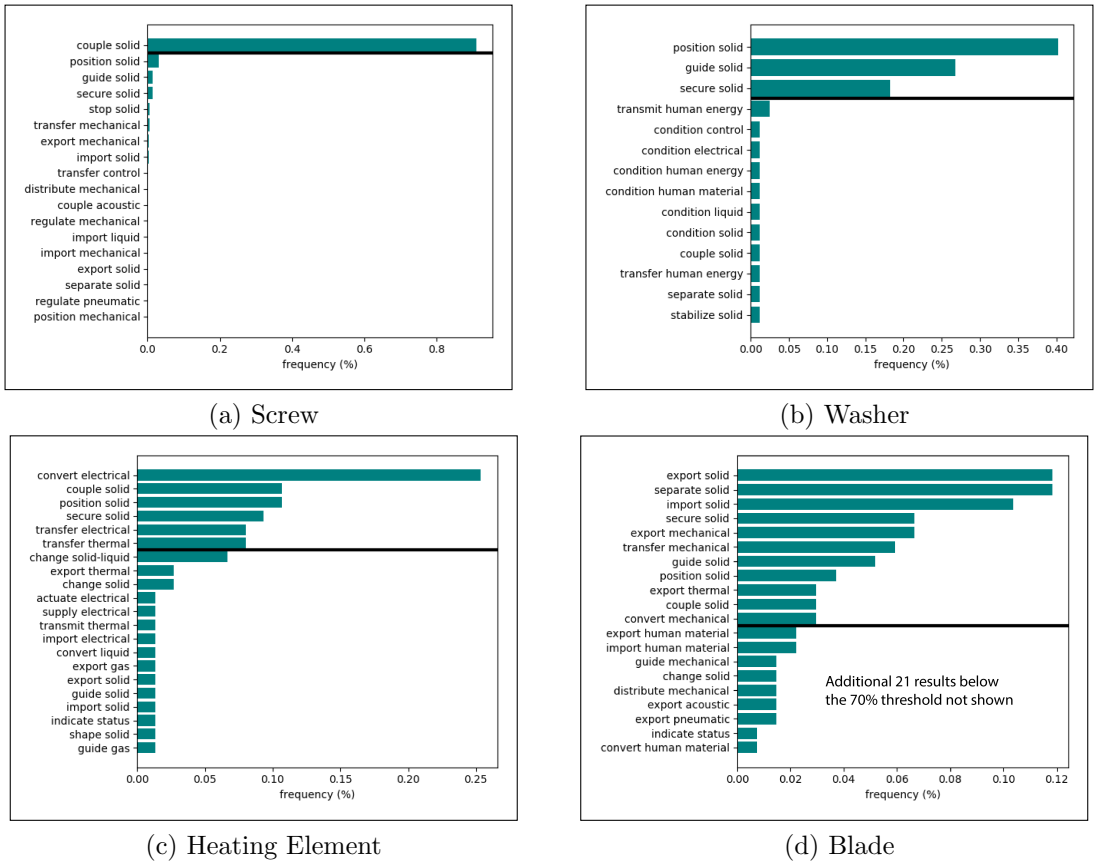
As seen in Figure 4.5 A, the component screw had one result above the threshold

because the frequency result for *couple solid* is 91%, the remaining 17 functions and flows only contribute to 9% of the results . While screw only has one result, blade (Figure 4.5 D) is representative of a component with more function and flows returning 11 results above threshold, an additional 21 results below threshold were not shown, for clarity in the figure. Washer and heating element can also be seen in Figure 4.5. Additionally, components with the most results were reservoir, circuit board and wheel with 22, 20, and 16 function and flow combinations in threshold respectively. We found that 98% of the dataset has at least 2 or more CFF combinations per component.

Frequency is calculated as the ratio of the number of times the CFF combination occurs over the total number of CFF combinations for that component. Returning to the screw example, the function and flow *screw couple solid* occurs the most at 589 times out of a total of 647. Conversely the consumer products dataset had 200 CFF combinations that only occurred once, which returns a ratio of 1/1 or 100% frequency. The other four datasets followed this same trend with a larger percentage of results occurring once or twice and a lower percentage occurring more than 20 times. As we expand the data in the repository we hope to decrease the number of times a CFF combination occurs only once.

4.5.3 Using F1 Scores to Validate Accuracy

We used the 10-fold cross validation method to quantify the accuracy with precision, recall, and the F1 score when applying the top 70% of the most frequent



(a) Screw (b) Washer

(c) Heating Element (d) Blade

Figure 4.5: Frequency Algorithm Results for Components in the Consumer Products Dataset

functions and flows found for a component for each of the three testing datasets. The number of products in each dataset, the size of a single fold (which is also the size of a testing set), and the size of the remaining nine folds (the size of the training set) is show in Table 4.4.

Each training dataset is tested against three testing datasets, itself, consumer products, and Black and Decker products. The results of the average F1 scores are shown in Table 4.5. For each of the three testing datasets, we performed a

Table 4.4: Sizes of Testing and Training Sets

Dataset	**Number of Products**	**Size of Testing Set (Size of Single Fold)**	**Size of Training Set (Size of Nine Folds)**
Blade	32	3	29
Heating Element	23	2	21
Reservoir	44	4	40
Consumer	142		
Black and Decker	12		

single factor ANOVA test to see if there is a significant difference in our F1 scores across the training datasets. We found that all three testing datasets **(blade, reservoir/container, and heating element)** were significantly different with =0.05. We then performed a two-sample t-Test assuming equal variances to determine within each testing dataset, which training sets were significantly different. Within each testing dataset, there was a significant difference found for all combinations, except the comparison of **Blade and Black and Decker** within the **Blade** testing set. The direction of the significant difference trends toward the consumer products dataset, which consistently had the highest F1 score.

Table 4.5: F1 Scores

Testing dataset: Blade		**Testing dataset: Reservoir**		**Testing dataset: Heating Element**	
Training dataset	**F1 scores**	**Training dataset**	**F1 scores**	**Training dataset**	**F1 scores**
Blade	0.4354	Reservoir	0.3967	Heating Element	0.4044
Consumer	0.4471	Consumer	0.4072	Consumer	0.4067
B&D	0.4419	B&D	0.2937	B&D	0.3215

4.5.4 Linear Functional Chains

In this section, we test the automation process described in the methods by building likely functional chains for four single components, which are the same four

components featured in section 4.2, Figure 4.5. The results of the functional linear chains can be seen in Figure 4.6. Screw is a very simple example with only one function and flow. As demonstrated in the results, components vary in complexity and therefore vary in functional chains. This complexity is based both on the component itself, such as the difference between screw and blade, but is also based on the product in which the component performs the function, for example blade within a knife versus blade within a more complex product like a jigsaw. We apply the following grammar rules adapted from Bohm and Stone, to the blade functional chain; a) import is automatically placed as the first function for a chain and b) export is automatically placed as the last function for a chain [4]. The grammar rules also dictate that the convert function has separate inflows and outflows; therefore the automation would branch off the function-flow export thermal from convert mechanical. This same rule is applied to the results of heating element, convert electrical is branched off into transfer thermal.

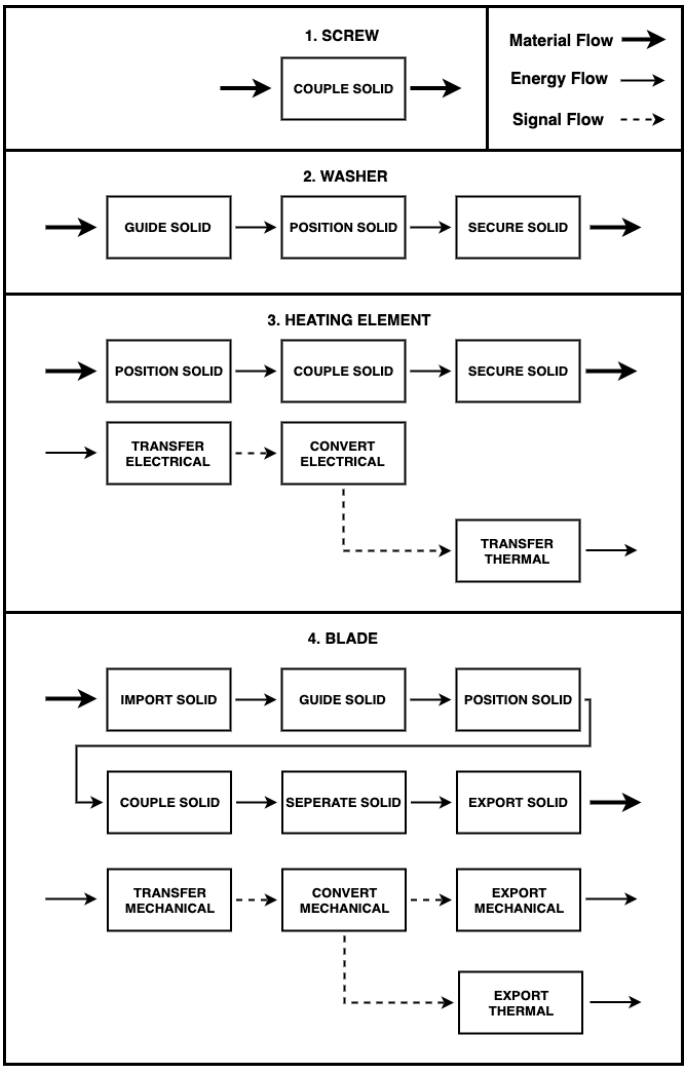


Figure 4.6: Component Based Linear Functional Chains

4.6 Discussion

In review, we mined the Design Repository for CFF combinations, we then applied a Pareto optimal threshold to find the most likely combinations, developed linear functional chains for individual components, and by validating the accuracy of the

frequency calculation and thresholding algorithm, we were able to test our hypothesis. We hypothesized that restricting the training set to constitute products that all share a similar component would give more accurate results for automating the generation of linear functional chains. For example, products having the component blade would have more similar functionality with other products having the component blade as opposed to products outside that dataset. As stated before, we found support with this hypothesis in previous work, using one product, the Delta jigsaw, as a validation method [44].

In this research, we had four general findings: **Finding 1:** With more robust validation methods, the results in Table 4.5 show that *learning from the most possible products will return a higher accuracy than any restricted-size dataset.* The component-specific datasets had lower accuracy when cross-validated against component-specific data than when cross-validated against all consumer products. The Black and Decker dataset is the smallest, containing 12 products, and consistently had the lowest F1 score when used as the training set. The consumer products dataset is the largest, containing 142 products, and consistently had the highest F1 scores.

Finding 2: We suggest that because the F1 score is calculated for an entire testing set, which often contains rare components that might have only one function and flow in the testing set, this may decrease the overall accuracy of function and flow results per component. As is often the case in large datasets, the accuracy of the data input can be a concern. Over the 20 years of the development of the Design Repository, many different contributors have worked on this project.

This turnover has led to some inconsistencies in the data; for example, container and reservoir are often used interchangeably or as seen in Figure 4.5, *screw* is 91% correlated with *couple solid* but there are 17 other results, which could be due to individual input variations. *This noise of the additional rare or mislabeled CFF combinations in the datasets can certainly reduce the accuracy of the results, especially for the larger consumer products dataset.*

Finding 3: While finding 1 suggests that learning from more data returns more accurate results, restricting the dataset based on the component may return more refined results for functionality. For example, the heating element, and reservoir/container component-specific datasets have six CFF combinations for the component *heating element*, the consumer products dataset has 10, and the blade dataset only returned one result. Heating element and reservoir/container have a high overlap in products, such as coffee makers, but blade products are unlikely to contain heating element as a component. *There may be times when a designer desires more refined results and a smaller learning dataset can be used if the products have the component of interest in the learning set.*

Finding 4: In developing the linear functional chains, we demonstrated more simple examples, such as *screw* and *washer*. As complexity increases, grammar rules are necessary to order the function and flow results. Only two existing grammar rules applied to our findings in heating element and blade. *As we expand our work in developing linear functional chains, we will need to expand on the research around grammar rules to create additional rules required to connect flows at the interface of components.* Individual analysis allows for the development of new

rules to handle each situation, but automation is possible based on investigating the interactions between component, function, and flow. While significant future work is required to fully automate the functional modeling of a product, these findings offer a starting point

4.7 Conclusion

Functional modeling is a complicated and challenging process for both novice and expert designers. However, during concept generation in product design, it is imperative to derive the functionality of the intended product because the function of the product is critical in linking customer needs to a form solution. We know from research, designers often think and design in component-based solutions. Our research finds connections between component and function and flow, information that can provide the designer with the functional breakdown of components. A functional approach to design is specialized; functional design accounts for variance in design for different purposes. We used data from the Design Repository to find the CFF combinations for five datasets: all consumer products, a Black and Decker consumer product family, and consumer products with the component blade, heating element, and reservoir. Our automation algorithm orders the frequencies from largest to smallest, sums the frequencies of each CFF combination, and uses a threshold of 70% of the sum of frequencies of combinations for each component. This threshold is the point where most of the functionality is preserved with a minimal contribution of error.

We then applied existing grammar rules to create component-based linear functional chains, the first step in automating functional modeling. Our results confirm the notion that function and flow correlations can be used to build a linear functional chain of individual components within a product. We found that the accuracy of data mining depends on the size and quality of the learning set used, with larger datasets providing more accurate results. However, using a broad or narrow dataset will depend on the goals of the designer.

Research has found inconsistency among designers when building functional models. We think that our future work towards an automated functional model generator will ultimately help standardize the language and syntax used in functional models, just as the work on Functional Basis and Component Basis terms have helped improve language and syntax consistency in the repository. As we have seen in the data in the design repository, designers are individuals dealing with human bias and perceptions; automation can help create more uniform functional models.

This uniformity will improve the process of designers contributing to the design repository, and enable more products to be added with higher consistency. The repository provides the user with a wealth of information; however, in its current form, the repository can be challenging to navigate for novice users. As stated above, streamlining the process of adding new products with automating functional modeling allows not only individual products to be added by users but also the addition of entire repositories. Enabling products to be entered by users will further increase the size and quality of the data in the Design Repository and increase the

accuracy of our automation process. This automation will also allow engineers to design a new product based on components and receive the functionality of the components.

Chapter 5: A Weighted Confidence Metric to Improve Automated Functional Modeling

Authors

Katherine Edmonds Design Engineering Laboratory
School of Mechanical, Industrial &
Manufacturing Engineering
Oregon State University
Corvallis, Oregon 97333
Email: edmondka@oregonstate.edu

Alex Mikes
Design Engineering Laboratory
Email: mikesa@oregonstate.edu

Bryony DuPont
Design Engineering Laboratory
Email: Bryony.DuPont@oregonstate.edu

Robert B. Stone

Design Engineering Laboratory

Email: Rob.Stone@oregonstate.edu

Proceedings of the 2020 ASME International Design & Engineering Technical Conferences

46th Design Automation Conference (DAC)

IDETC 2020

August 16-19, 2020, St. Louis, MO, United States of America

5.1 Abstract

Expanding on previous work of automating functional modeling, we have developed a more informed automation approach by assigning a weighted confidence metric to the wide variety of data in a design repository. Our work focuses on automating what we call linear functional chains, which are a component-based section of a full functional model. We mine the Design Repository to find correlations between component and function and flow. The automation algorithm we developed organizes these connections by component-function-flow frequency (CFF frequency), thus allowing the creation of linear functional chains. In previous work, we found that CFF frequency is the best metric in formulating the linear functional chain for an individual component; however, we found that this metric did not account for prevalence and consistency in the Design Repository data. To better understand our data, we developed a new metric, which we refer to as weighted confidence, to provide insight on the fidelity of the data, calculated by taking the harmonic mean of two metrics we extracted from our data, prevalence, and consistency. This method could be applied to any dataset with a wide range of individual occurrences. The contribution of this research is not to replace CFF frequency as a method of finding the most likely component-function-flow correlations but to improve the reliability of the automation results by providing additional information from the weighted confidence metric. Improving these automation results, allows us to further our ultimate objective of this research, which is to enable designers to automatically generate functional models for a product

given constituent components.

5.2 Introduction

The data stored in design repositories is useful to designers during the concept generation phase, particularly for design activities such as generating functional models. The Design Repository [1], unique in the depth and breadth of information and abstraction, is a product database where data can be searched and retrieved at different levels of abstraction including the functions and flows associated with the constituent components of each product[43]. However, our previous research with the Design Repository discovered that there are outliers in the product function data that are either inconsistent or rare in occurrence. We developed a metric to consider the fidelity of this data and allow designers to retrieve the data still, yet be aware of the fact that the data may be an anomaly. Frequently outliers in data are discarded or not included in the analysis by the researchers in order to reduce noise. However, in the concept generation process, designers may receive valuable creative insight from these unlikely results.

The concept generation phase of the design process is an area where the most creativity and innovation occur [47]. This early design phase is where the least cost for changes occurs, so there is significant research on tools and tactics to improve the concept generation phase and how to improve the efficiency of the process. There are multiple tools that product designers use during this phase

[1]The Design Repository is a database of design information. It is currently housed at Oregon State University. A basic web interface is available at ftest.mime.oregonstate.edu/repo/browse

to help guide the creative process towards a viable concept. One of these tools is deriving the functionality of the product through a functional decomposition, graphically represented by a functional model [45] [2]. Consistency and uniformity are built into the process with the widely accepted use of the Functional Basis and Component Basis terms [40] [17] [23]. However, even with consistency built into the process, functional models can vary widely by the individual user input [29]. Additionally, functional modeling if often overlooked or omitted from concept generation because designers have a difficult time considering design in terms of the functionality of the product. Rather designers are more comfortable with component-based solutions, often benchmarking existing products [28]. However, research has shown that concept generation is more robust when the function is considered [45]. Incorporating functional modeling early into the design phase can help the shift of resources in the project lifecycle to earlier in the design process when the cost of making changes is low, but the impact of those changes is high.

With the knowledge of the importance of incorporating functional decomposition into the early design phase, we have focused our research on how to improve the process of developing functional models with the use of existing product functionality data from the Design Repository. Using the existing connections between component function and flow from the Design Repository, we are mining data to work towards automating functional modeling. Our research team's reasoning for automating functional modeling is three-fold, increasing the use and comprehension of functional modeling, improving the Design Repository by expanding the data and streamlining the process of adding products, and connecting components

to function and flow to allow for the inclusion of function in component-based design.

We are building on our research team's previous work towards the automation of functional modeling and the expansion of the Design Repository. Utilizing the information in the Design Repository, this work is centralized around finding the correlations between components and function and flow or **CFF combinations**. While this work will be described more in-depth in the *Background* section, a brief introduction follows here.

We begin by expanding the *Form Follows Form* approach, which is based on the concept that designers most often think in terms of components rather than function when working in the concept generation phase [5]. Bohm et al. calculated the CFF frequency of the function and flows correlated with each component *separately*. We first used the Apriori algorithm to find the *combined* associations between component and function-flow using a subset of the consumer products data, applying a threshold to determine the most likely functions and flows per component [44]. During data analysis, we found that some of the metrics of association rules were unnecessary. The team then simplified our calculations, focusing on the CFF frequency of CFF combinations, which is numerically equivalent to the confidence metric from association rules[12]. We developed an automation algorithm, referred to as the Automated Frequency Calculation and Thresholding Algorithm or AFCT, that returned the CFF frequency of the component-function-flow (CFF) combinations and applied a threshold the returned only the top 70% of functions and flows per component. We validated the accuracy of our algorithm on multiple

subsets of the consumer product dataset, finding that increasing the size of the dataset for data mining increases the accuracy of our automation algorithm[12]. Restricting the dataset essentially reduced the size of the results from which the algorithm could learn. The limitation of our current automation process is that *prevalence*, the measure of the commonness of the component, and *consistency*, the measure of how uniform the CFF combinations are per component, are not considered, which we refer to broadly as data fidelity.

The weighted confidence metric replaces a common approach of removing rare data; instead, our metric allows all data to be included by describing the data fidelity. We were unable to find a numerical tool or quantification that returned the synthesis of prevalence and consistency in our dataset, so we developed our own metric. Here we create a metric to account for prevalence and consistency that will be a better measure of confidence in the automation results than simple CFF frequency.

Our immediate research objectives are to 1) mine the Design Repository for the consumer product dataset, 2) apply the automation algorithm to calculate the frequencies of CFF combinations and apply the classification threshold, 3) develop a metric that would give more confidence in the automated results of our algorithm, and 4) test our methodology by developing example linear functional chains.

5.3 Background

5.3.1 Automated Frequency Calculation and Thresholding (AFCT) Algorithm

The Design Repository is the ongoing result of decades of repository research and is comprised of 142 consumer-based electro-mechanical products housed online through the Design Engineering Lab at Oregon State University [43, 7, 8, 22, 38]. Each product is divided into seven main categories of design information: artifact, function, failure, physical, performance, sensory, and media-related information types [6]. A visual reference of the data schema (i.e., the connections between data) is shown in Figure 5.1. These different levels of abstraction to help improve design knowledge and data-driven design decisions[43]. We define data-driven design as methodologies for extracting information and insights from data and existing research to improve design processes [34].

Our data-driven design approach focuses on a specific connection, the component-function-flow combination (**CFF combination**), by extracting the connection from the data in the Design Repository. In Figure 5.1, the letter B denotes the component basis type and function flow connection, and the letter A denotes the larger component-function-flow structure. The term artifact refers to the common component name, where the component basis term refers to the Component Basis terms developed by Kurtoglu et al. 2005 [22]. The function and flow utilize the Functional Basis terms developed by Stone and Hirtz [40] [17]. The Component

and Functional Basis terms allow us to compare CFF combinations to each other with the knowledge that there is consistency in the language.

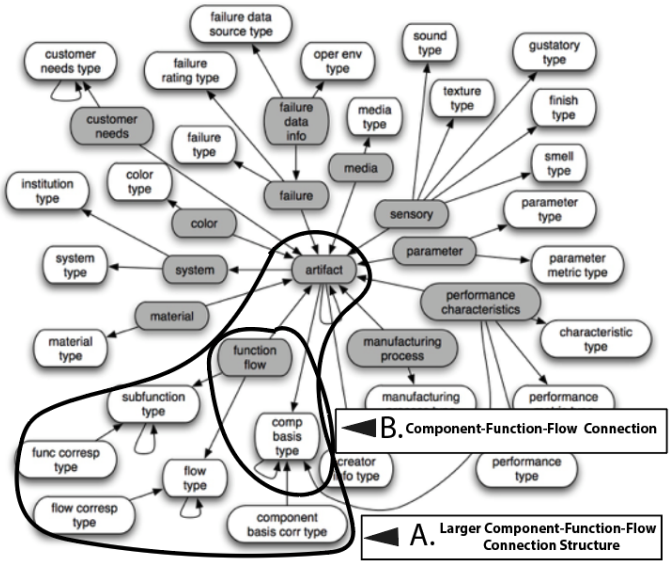


Figure 5.1: Design Repository Data Schema [6]

We pick up where the *Form Follows Form* (FFF) approach left off, working to capture the underlying functionality of the chosen components using data from the design repository [5][4]. In the FFF approach, Bohm et al. calculated the CFF frequency of function and flow associated with components *seperately*. Our research continues to build on this concept, attempting to streamline the automation process by *combining* the component-function-flow association refereed to as **CFF combinations**. We chose only to consider the incoming flows to simplify our analysis, as we found in analyzing our datasets that less than 5% of the results have different inflow and outflow. We first used association rules with the Apriori algorithm to find the CFF combinations using a small subset of the consumer

products data, applying a threshold to determine the most likely functions and flows per component [44]. Association rules are a type of data mining that describes the relationship between items in item sets [30] [25]. During data analysis, we found that association rules returned more metrics than we needed [12]. We simplified our calculations, focusing on the frequency (*CFF frequency* for clarity) of CFF combinations, which is numerically equivalent to the confidence metric from association rules. CFF frequency is calculated as the ratio of the number of times the CFF combination occurs over the total number of CFF combinations for that component. An example with the component *screw*, demonstrates the ratio. The function and flow *screw couple solid* occurs the most at 589 times out of a total of 647, so the CFF frequency of the combination is 589/647 or 91%. Some CFF combinations only occurred once in the dataset, which returns a ratio of 1/1 or 100% CFF frequency.

Next, we determined that a threshold needed to be applied to extract the most likely functions and flows for each component. The Pareto Frontier motivates the 70% threshold from the Form Follows Form method [4]. We found in our data analysis that the 70% threshold is often the point where adding additional functions and flows for a component contributed a negligible change in the sum of frequencies and decreased the accuracy of the automation results. We created an automation algorithm to report the likely functions and flows automatically; this algorithm is referred to as the Automated Frequency Calculation and Thresholding Algorithm or AFCT. The AFCT algorithm orders the CFF frequencies of the CFF combinations per component from largest to smallest, sums the frequencies of each

CFF combination, and then applied a threshold the returned only the top 70% of functions and flows per component. Edmonds et al. validated the accuracy of the AFCT algorithm on multiple subsets of the consumer product dataset, determining that the largest dataset was the most accurate [12]. This finding indicates that a restricted dataset limits the results from which the AFCT algorithm could learn.

The ultimate goal of this research is utilize data from the Design Repository to further the automation of functional models. A functional model is the graphical representation of the functional decomposition of a product, and an example of a Black and Decker Dustbuster can be seen Figure A.1 in the Appendix. Figure A.1 demonstrates the complexity of functional models. To simplify the process of automation, we begin by building individual-component-based linear functional chains. We have shown in previous research that finding associations between functions and flows, and components, allows us to build these linear functional chains [44] [12]. Starting with a simplified model, we can work out the issues and problems with automation rather than starting with such complexity as a full product functional model. With the CFF combinations returned from the AFCT algorithm, we build linear functional chains for components.

Functional decomposition has been the subject of extensive research [45] [2]. Some of this research involves developing grammar rules to help solve the consistency issue with building functional models [29] [38]. Kurfman et al. found that despite a formal language, repeatability was a challenge among both novices and experts [20]. These grammar rules help determine the appropriate order of the functions and flows for a product while developing a functional model. We apply

grammar rules to the creation of linear functional chains.

5.3.2 Weighted Confidence Metric

Weighting is a commonly used tool when dealing with statistical probabilities or uncertainty [46] [41]. Based on the idea that not all results are equal, a weight can be assigned to a probability to increase or decrease its influence on the results. In our work, the inequality in results comes from varying frequencies and consistencies in our data. In using our ACFT algorithm described above, we found that CFF frequency did not account for the prevalence or consistency of the CFF combinations in the dataset. In other words, a CFF combination that occurred five times could have the same CFF frequency as a CFF combination that occurred 500 times in the dataset. Weighting these rare CFF combinations the same as combinations with high prevalence can create a false sense of confidence in the analysis. Additionally, some CFF combinations only occur once, returning a CFF frequency of 100%. This data is likely associated with a component that does not often occur in products. However, data that has low prevalence is still useful and vital to include in our automation process. We do not want to eliminate the results with low frequency or consistency, but we want to indicate additional information about the influence that is not found in those metrics alone. Therefore, we developed quantitative descriptors for our data with the aim of using them to build an improved metric for CFF frequency.

OHalloran et al. developed a frequency weighting metric that helps understand

reliability and uncertainty in early design phases [33]. Their work uses The Design Repository to calculate and predict failure based on the number of occurrences. They calculate frequency weights and apply them to a Hierarchical Bayes model in a similar manner to the Holt-Winter method that is used to forecast based on Exponentially Weighted Moving Averages (EWMA) [18][35]. Their overall method is the Early Design Reliability Prediction Method (EDRPM), and it calculates weights based on occurrence instead of the time series data in EWMA [32].

Our method of calculating a weighted confidence metric is similar to the EDRPM because it accounts for occurrence (prevalence) and is similar to IPW because it accounts for rarity (consistency). We blend the two metrics together to give a weighted confidence factor that best represents the data in the Design Repository. We chose to use the harmonic mean to combine prevalence and consistency to create the weighted confidence metric because of its superior application in using ratios [13].

5.4 Methods

The purpose of this methodology is to develop a way to improve the CFF frequency data fidelity, ultimately improving our linear functional chain automation results. Below, we present the methods in four steps: 1) retrieve consumer products data from the Design Repository, 2) apply the CFF frequency and thresholding automation algorithm, 3) develop a weighted confidence metric, and 4) create linear functional chains.

Table 5.1: Metrics Developed For Weighted Confidence

Metric	Measure	Description	Example Component: Electric Wire	Example Component: Housing
CFF count per component		The number of CFF combinations per component in the dataset.	651	1257
Max CFF count per component		The component with the max number of CFF combinations in dataset.	1257	1257
Unique CFF combinations		The number of unique CFF combinations per component	39	101
Unique CFF combinations in Threshold		The number of unique CFF combinations per component within the 70% threshold of the dataset.	2	7
Prevalence	This metric accounts for the commonness of the component in the dataset	The ratio of the number of times a component occurs in the dataset to the max number of times any component occurs in the dataset	0.51	1
Consistency	This metric determines how uniform the CFF combinations are per component	The ratio of the total unique CFF combinations per component to the unique CFF combinations in the threshold dataset (scaled 0 to 1)	1	0.73
Weighted Confidence	This metric describes the both prevalence and consistency of the CFF combination data.	The harmonic mean of prevalence and consistency	0.68	0.85

Step 1. Retrieve Data

To test this methodology, we chose to work with the largest dataset in the Design Repository-the consumer products dataset. We previously found the consumer products dataset is the most accurate and gives the most confidence in the automation results. In verifying the accuracy of the AFCT algorithm results, we tested the algorithm on four smaller datasets, both component-specific and a company based product portfolio. We found that learning from the most possible products returns the highest accuracy [12]. To extract the information needed, we query the Design Repository for the component and function and flow connection for the 142 consumer products.

Step 2. Apply the Automated Frequency Calculation and Thresholding (AFCT) Algorithm

We utilize the automation frequency calculation and thresholding algorithm (AFCT) developed previously to retrieve CFF frequency and thresholding data for the consumer products dataset [12]. Once the threshold was applied, the unique function and flows per component were reduced to a range of 1 to 22 compared to 1 to 101.

An example component can be seen in Figure 5.2. For the component *pulley*, the CFF frequency of the first two functions and flows sums to 63%, so the third is added to the list to reach the 70% threshold. This brings the sum to 74% and results in rejecting the last four results. For example, based on the results from Figure 5.2, the order of the linear functional model would be *secure solid, guide solid*, and *transfer mechanical*. We use grammar rules and design knowledge to put the function and flow in linear order, not the magnitude of the CFF frequency. Note that the third and fourth results for *pulley* have the same CFF frequency, the AFCT algorithm arbitrarily removes one of these results over threshold. This limitation will be discussed in more depth in the *Assumptions and Limitations* section.

Step 3. Develop a Weighted Confidence Metric

As described previously, the AFCT algorithm returns the most likely functions and flows per component. While the results of the algorithm are invaluable for the au-

Pulley	CFF Frequency	Running sum of CFF frequency	Threshold
Transfer Mechanical	0.37	0.37	Keep
Guide Solid	0.26	0.63	Keep
Secure Solid	0.11	0.74	Keep
Change Mechanical	0.11	0.85	Reject
Position solid	0.05	0.90	Reject
Convert Human Energy	0.05	0.95	Reject
Couple Solid	0.05	1	Reject

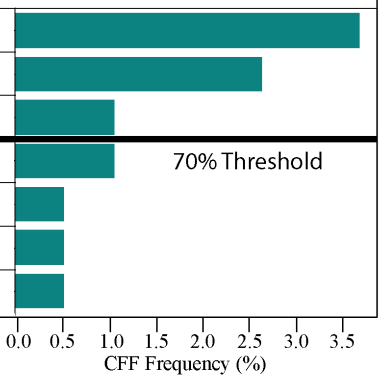


Figure 5.2: Example To Illustrate Threshold Automation For The Component Pulley

tomation process, the CFF frequency calculation does not indicate the prevalence or consistency of the component. For example, *housing, electrical cord*, and *screw* were three components that appeared well over 100 times in the repository. With examples like this, we can be confident in the fidelity of AFCT algorithm results. However, many results only occur once in our dataset, returning a 100% CFF frequency. These rare CFF combinations have a high CFF frequency, yet the fidelity of this result is much lower. This example demonstrates that the magnitude of the CFF frequency is not indicative of the fidelity of the data. To improve the fidelity of the results of our AFCT algorithm, we developed a weighted confidence metric to account for the data fidelity of the automation results.

In order to create the weighted confidence metric, we took the harmonic mean of two metrics, *prevalence*, and *consistency*. To reiterate, *prevalence* measures

the commonness of the component, and *consistency* measures how uniform the CFF combinations are per component. These metrics are described in Table 5.1. Two example components demonstrate the numbers used to calculate the weighted confidence metric and are shown in Table 5.1. Consistency was scaled from 1 to 0 to make it equal in magnitude to prevalence so that the harmonic mean could be estimated. Harmonic mean in Equation 5.1 is a more representative mean when dealing with ratios rather than the arithmetic mean [13], where n is the number of variables used to calculate the mean, in our case $n = 2$ (consistency and prevalence), a_1 is consistency, and a_2 is prevalence.

$$HarmonicMean = H = \frac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_n}} \tag{5.1}$$

High prevalence is demonstrated in Table 5.1, the component *housing* occurs 1257 times in the dataset. *Housing* is a component that occurs in almost all consumer products, so naturally, it would have a high prevalence. An example of low prevalence is *analog display*, which only occurs once, indicating that only one product in the repository has this component. An example of high consistency is shown in Table 5.1, *electric wire* has the highest ratio of total unique CFF combinations to unique CFF combinations in threshold, 39/2. Demonstrating that even though there are 39 combinations for electric wire, only two of those combinations represent 70% of the results, *transfer electrical* (44%) and *couple solid* (26%).

Step 4. Create Linear Functional Chains

We can use the likely functions and flows found by the AFCT algorithm to develop linear functional chains. The weighted confidence metric can be used to determine the fidelity of the linear functional chain. We show four distinct example components from the dataset that are representative of the four combinations of high- and low-CFF frequency and weighted confidence below:

1. High CFF frequency, high weighted confidence
2. Low CFF frequency, high weighted confidence
3. Low CFF frequency, low weighted confidence
4. High CFF frequency, low weighted confidence.

These categories represent the four quadrants in Table 5.2. As stated above, CFF frequency alone cannot determine the prevalence and consistency of data. Combining the weighted confidence metric with CFF frequency, as seen in Table 5.2, provides additional information improving our automation process. High CFF frequency indicates that the component has few functions and flows associated

Table 5.2: Description Of The Combination Of CFF Frequency and Weighted Confidence

High Weighted Confidence	Multiple results per component that occur many times.	One or two CFF results that occur many times in the dataset.
Low Weighted Confidence	Multiple results per component that only occur a few times.	One or Two CFF results per component that only occur once.
	Low CFF frequency	**High CFF frequency**

with it, and high weighted confidence indicates that the component often appears in the data and has consistent, unique function and flows. Low CFF frequency indicates that the component has many associated functions and flows, while low weighted confidence indicates that the component is rare in the data and is not consistent with unique function and flows. We chose an example component from each quadrant to demonstrate an automated linear function chain, using the function and flow combinations found by the CFF frequency calculation and thresholding algorithm.

To form the order of the linear functional chain, if there are more than one function and flow per component, we order the functions and flows based on previously created grammar rules. For example, Bohm et al. state that the import function occurs first and only once per flow in a chain of components, and that export is the last function in a chain of components [4]. Grammar rules do not exist for every combination of function and flow. Currently, we are creating the linear functional chains by hand using expert knowledge. As this research progresses, we will develop additional grammar rules, which will help continue to automate the process of developing functional models.

ASSUMPTIONS AND LIMITATIONS

The primary limitation in our previous work (that this research seeks to eliminate) is that a CFF combination could appear a few times or several hundred times in the dataset, and with only the CFF frequency calculation, there was not a way

to determine the difference. We make key assumptions in this research: primarily, we assume that due to the use of the Functional and Component Basis terms, the data in the Design Repository is consistent. For example, one function and flow combination that appeared for both components *rivet* and *screw* is *couple solid*. This consistency allows us to compare function and flow across components. However, we know that at times due to multiple entries from different researchers, we do need to account for variance and error, such as the component basis terms *container* and *reservoir* being used interchangeably. Input fidelity and linguistic imprecision, such as the difference between container and reservoir, are two concerns. This is ultimately why we chose to develop the weighted confidence metric to help account for any erroneous data.

While we found the 70% threshold worked for the majority of components, some components fall outside this typical pattern. For example, the components *condenser* and *screen* have an even split of the CFF frequency across all results. This equal distribution creates a unique situation where the threshold arbitrarily eliminates the last function and flow. Figure B.1 in the Appendix shows the AFCT algorithm results for both components. In cases like this example and the *pulley* example (Figure 5.2), future work is needed to optimize the threshold in the AFCT algorithm.

5.5 Results and Discussion

5.5.1 Automated CFF frequency Calculation and Thresholding (AFCT) Algorithm

The 142 products were composed of 132 different component basis types and 161 functions and flows that were combined to create the CFF combinations. The query and algorithm returned 11,394 CFF combinations for the 142 consumer products in the Design Repository. The range, average, and median of the different CFF combinations can be seen in Table 5.3.

5.5.2 Weighted Confidence Metric

The weighted confidence metric improves the automated results of the CFF frequency calculation and thresholding algorithm by incorporating the prevalence and consistency of the data. The relationship between consistency and prevalence was not proved to be a 1-to-1 relationship for a significant portion of the data, demonstrating the importance of including both metrics in the weighted confidence calculation, see Figure 5.3 trend line.

Figure 5.4 shows the relationship between CFF frequency and weighted confidence. Each point represents one CFF combination. The size of the bubble is the number of CFF combination occurrences per component in the dataset; for example, *housing* has 1257 CFF combinations (the max number of occurrences for a component in the dataset) seen in the top left of the figure. Figure 5.4 shows

Table 5.3: Range, Average, and Median Of The Component Function Flow Associations

	Range	Average	Median
Total individual CFF combinations per component	1-1257	133	55
Individual CFF combinations per component within threshold	1-908	105	45
Total unique CFF combinations per component	1-101	27	22
Unique CFF combinations per component within threshold	1-22	10	9

that the weighted confidence metric is needed to improve data fidelity of CFF frequency results, as high weighted confidence values are found across all CFF frequency values. A low CFF frequency is not indicative of the importance of the CFF combination; rather, it simply indicates that there are multiple results per component. The average number of unique functions and flows per component is 10, illustrating that most components have multiple associated functions and flows. Figure 5.4 shows the large percentage of the CFF combinations have a CFF frequency below 40%. For example, *housing*, which has 7 CFF combinations in threshold, resulting in low CFF frequency for each combination. However, housing has the highest prevalence in the dataset, resulting in a high weighted confidence value, which is more indicative of the fidelity of the data than the CFF frequency values. In Figure 5.5, we have partitioned the parameter space into four quadrants to show examples of four combinations of CFF frequency and weighted confidence values discussed in the methods and shown in Table 5.2. The results demonstrate

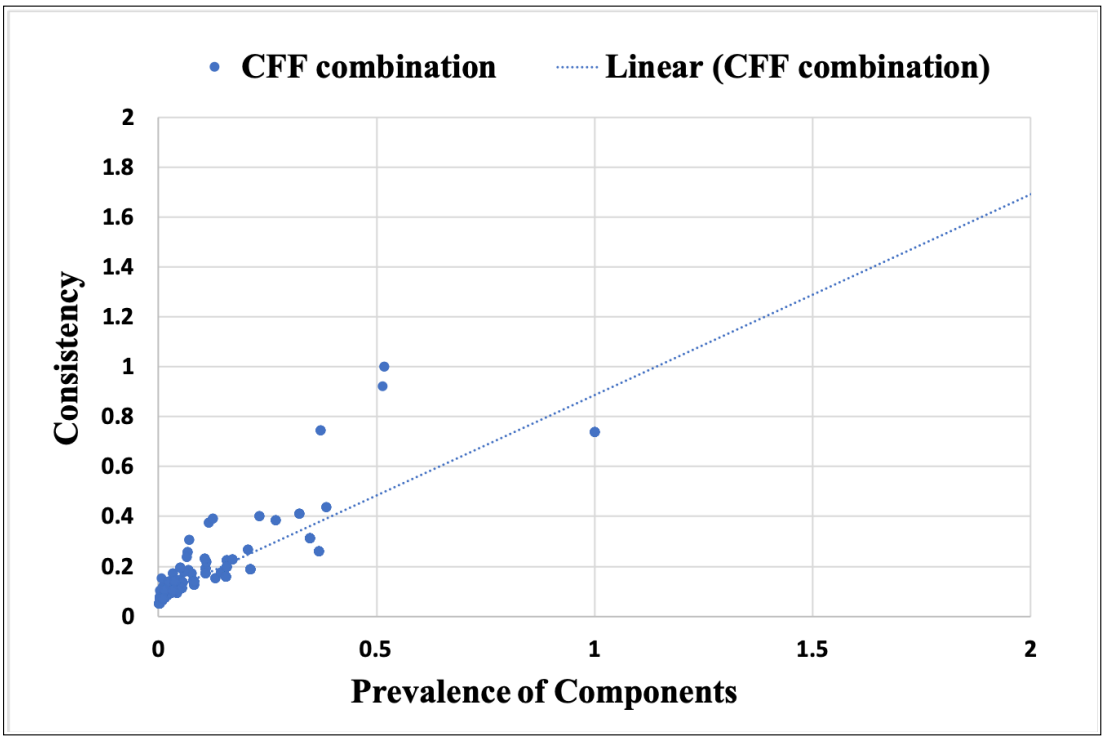


Figure 5.3: Consistency Versus Prevalence

that while CFF frequency is needed to return the likely functions and flows per component, the magnitude of CFF frequency is not essential; however, the magnitude of the weighted confidence can indicate confidence in the automation results.

Here, we briefly describe four specific results from Figure 5.5.

A. Low CFF frequency, high weighted confidence The automation algorithm returned 7 CFF combinations within the threshold for the component housing. Multiple CFF combinations per component result in a lower CFF frequency per combination. Housing is the component with the highest prevalence in the dataset at 1257, and it has high consistency with a ratio of 101 unique CFF com-

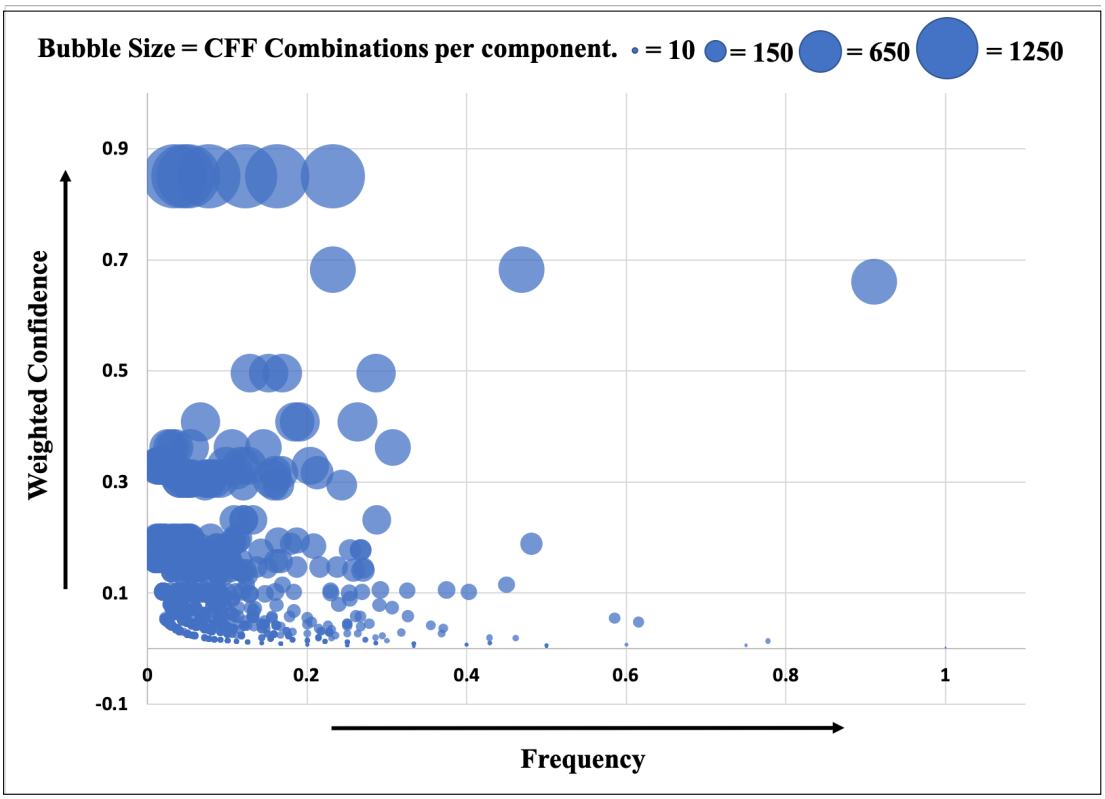


Figure 5.4: Weighted Confidence Verus CFF Frequency With The Occurrence Of The Component As The Size Of The Bubble

binations to 7 unique CFF combinations in the threshold. Since housing has both a high prevalence and consistency in the dataset the weighted confidence value is also high at 85%. The top CFF combination was *Housing - Position Solid* with a CFF frequency of 23%, the other 6 combinations had a lower CFF frequency.

B. High CFF frequency, high weighted confidence Screw has very high CFF frequency because within the threshold; there is only one CFF combination, *Screw - Couple Solid*. This combination has a CFF frequency of 92%, meaning *Couple Solid* is the most likely function and flow for the component *Screw*. Like housing,

screw has both a high prevalence and high consistency. The prevalence is how often screw appears in the dataset, 647 times. Consistency is the ratio of unique CFF combinations to unique CFF combinations within threshold, which is 18 to 1. The weighted confidence metric is 66%.

C. Low CFF frequency, low weighted confidence *Condenser- Convert Gas* is an example of a CFF combination that has low CFF frequency but also a low weighted confidence value. The automation algorithm returned five unique CFF combinations for condenser, but there were only a total of six results in the Repository. The component condenser only shows up in three of the 142 products in the Repository, indicating that this a rare component in our products. The low weighted confidence metric, 0.8%, indicates low fidelity of the automation results.

D. High CFF frequency, low weighted confidence *Analog Display-Indicate Mechanical* is an example of a CFF combination that only occurs once in the Repository. The CFF frequency is therefore very high, 100%, but the weighted confidence is very low, 0.15%. CFF combinations that only occur once return a false high CFF frequency that can be illuminated by the low weighted confidence metric.

5.5.3 Linear Functional Models

To translate our findings into automation, we developed four linear functional chains based on our four examples above in Figure 5.5. Each example component came from one of the four quadrants shown in Table 5.2. The linear functional

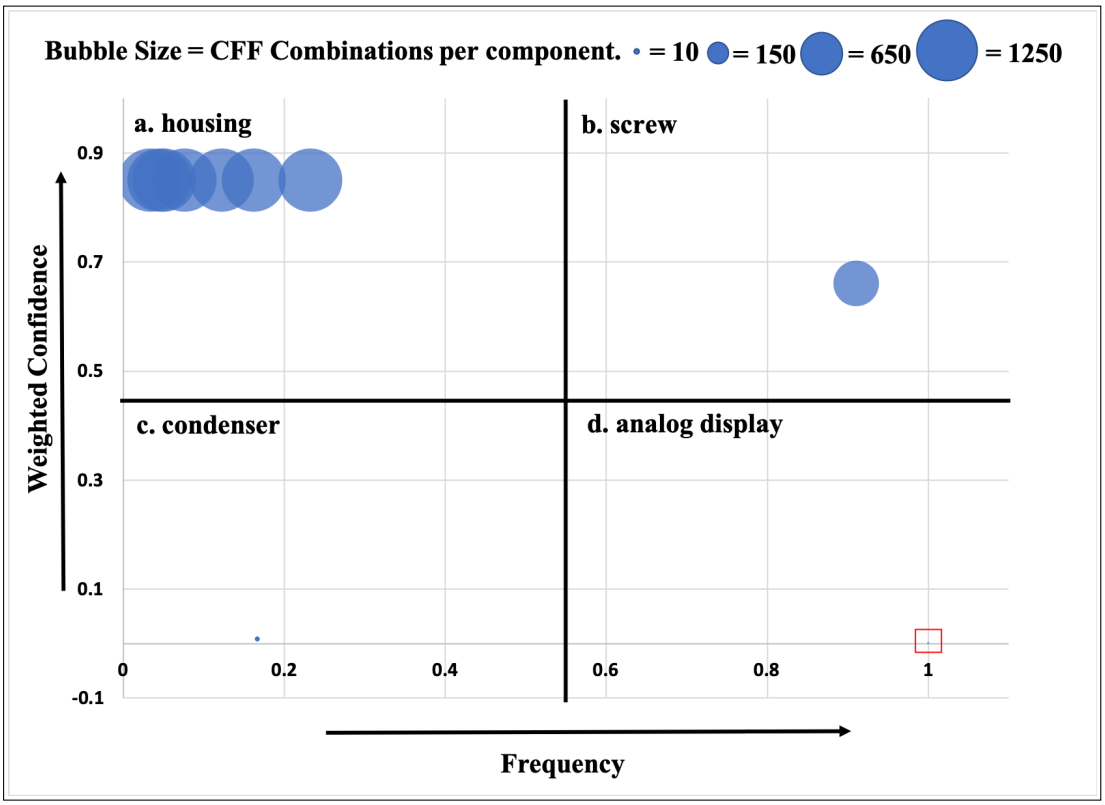


Figure 5.5: Example Data For The Four Quadrants Of Combined Weighted Confidence And CFF Frequency.

chains are a demonstration of the automation process described in the methods. The AFCT algorithm returns the most likely functions and flows for a component. If there is more than one function and flow returned for a component, the results are ordered using existing grammar rules and expert knowledge. The components in Figure 5.6 demonstrate that components vary in complexity and therefore vary in functional chains. Screw, for example, has only one function and flow, *couple solid*, whereas condenser has many more functions and flows. This complexity

can also be attributed to the function the component performs in the product; for example, a knife blade performs a more straightforward function than a jigsaw blade.

For the linear function chains shown in Figure 5.6, the higher weighted confidence metric for housing and screw indicates higher data fidelity than the two components with lower weighted confidence, condenser, and analog display. As seen in Figure 5.6 A., housing is an example of a component with multiple results; these results need to be ordered linearly. For the flow of human material, we apply the following grammar rules adapted from Bohm and Stone a) *import* is automatically placed as the first function for a chain and b) *export* is automatically placed as the last function for a chain [4]. Currently, grammar rules do not exist to describe the order functions such as *position, guide, couple, and secure*. Therefore, using our knowledge of functional models, we determined that *position solid* must come before *guide solid*, and *guide solid* would come before *couple solid*, and *couple solid* would come before *secure solid*. The same reasoning was applied to the component condenser, Figure5.6 C. As we move toward automation, we will continue to develop grammar rules to improve the machine learning of our process. The grammar rules also dictate that the convert function has separate inflows and outflows; therefore, the automation would place transfer gas before convert gas for the component condenser seen in Figure 5.6.

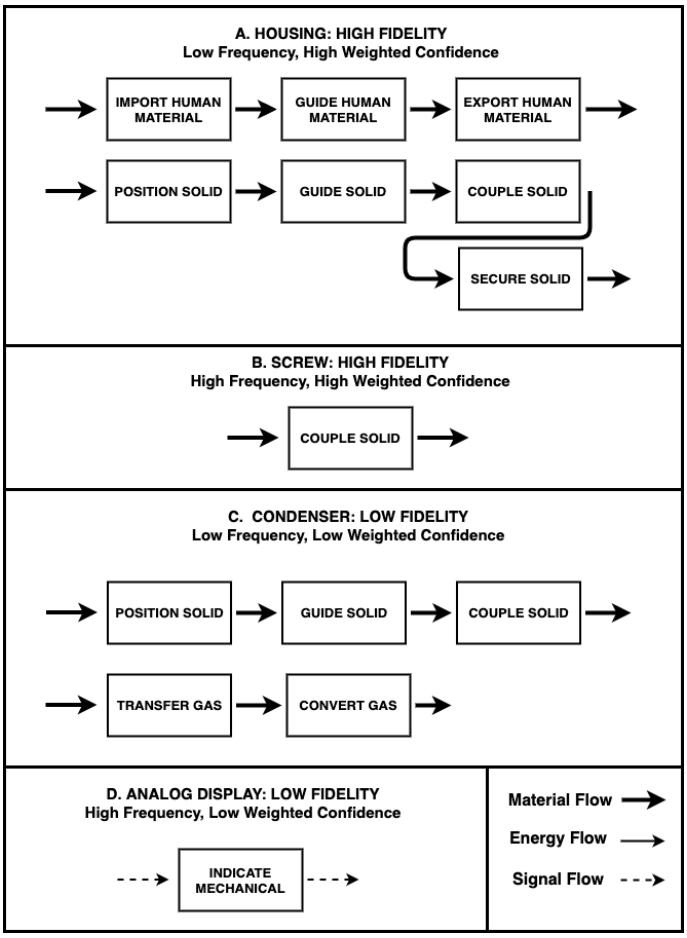


Figure 5.6: Linear Functional Chains Of The Four Examples in Figure 5.4

5.6 Conclusion

We set out to assess the prevalence and consistency of the outlying component-function-flow combination (CFF combinations) in the Design Repository data. In previous work, we found that CFF frequency was a suitable metric to determine a components likely function and flow, but was unable to identify the prevalence

and consistency of the components in the dataset. Our work developed a weighted confidence metric to supplement CFF frequency during the automation process. The weighted confidence metric supports CFF frequency by analyzing the components data fidelity, identifying the range of low to high confidence. Figure 5.4 demonstrated the distribution of CFF combinations across frequency and weighted confidence. The range of distribution of CFF frequency is shifted towards the lower end of the spectrum because the majority of components in the consumer products dataset have multiple function and flow outputs resulting in the division of CFF frequency across all instances. However, the weighted confidence is distributed more evenly across all CFF combinations, indicating that there is a range of data fidelity in the Design Repository. Ultimately we need both metrics in order to automate the process; the CFF frequency metric returns the most likely function and flow results for each component, and then the weighted confidence metric accounts for prevalence and consistency in the data. With the weighted confidence metric, we are now able to capture occurrences of all components in a given dataset, thus improving the results of our automation algorithm. By including the weighted confidence metric, we have eliminated the tendency to discard useful outliers to reduce the noise in analysis such that these outliers can now be included in the concept generation process. The inclusion of these outliers can provide valuable creative insight to designers. We have provided a simple method that researchers could implement with their own datasets to weight results versus discarding outliers, ultimately increasing the robustness of data analysis.

This methodology has helped increase the utility of automated functional mod-

eling. However, there is still much to be done to fully automate the process of creating complex functional models for entire products. The next step would be to integrate the weighted confidence metric into the AFCT algorithm, returning both metrics. Future work should also look at optimizing a threshold specific to each component, identifying where adding additional function-flow combinations has a negligible change. In order to broaden linear functional chains to the full functional model, work must be done on connecting components to each other, as well as connecting the components through flows.

One of the main goals of this research is to help expand the Design Repository. As we develop our automation process, it becomes easier in the future to add information from other repositories, which significantly expands our database. We are working with additional OSU researchers to house the information from an existing Sustainable Design Repository in the OSU Design Repository [14]. Combining this information adds additional products, as well as sustainable design information such as LCA analysis and manufacturing processes. Expanding on the work presented in the Function-Human Error Design Method (FHEDM), Soria et al. have been using Design Repository data to develop new relationships, such as incorporating the user, user interactions, human error [48] [37]. The database structure of the Design repository provides mapping and connections between categories of the product systems, expanding these connections to include sustainability and user-system interactions will bring these important considerations to the early phase of design decisions.

Chapter 6: Discussion

This research is the first step towards automating functional modeling forward. In review, we mined the data from the Design Repository for component-function-flow (CFF) combinations and developed linear functional chains based on individual components.

In the first manuscript, we first tested our hypothesis that restricting the training dataset to products that all share a similar component would give more accurate results for automating the generation of linear functional chains. For example, products having the component blade would have more similar functionality with other products having the component blade as opposed to products outside that dataset. We found support with this hypothesis in previous work, using only one product, the Delta jigsaw, as a validation method [44]. Expanding our datasets and validation methods, we tested the accuracy of the automated frequency calculation and thresholding (AFCT) algorithm. We expanded to five datasets, which included three component-based datasets (blade, heating element, and reservoir/container), one product portfolio (black and decker), and the full consumer products dataset. We used the k-fold cross-validation method to test the accuracy of our algorithm. We tested the three different component-based datasets against itself, the Black and Decker dataset, and all consumer products dataset. The results of this validation process determined that our hypothesis was false, the consumer reports

dataset consistently had the highest F1 score, indicating that the accuracy of data mining does depend on the size and quality of the learning set used, with the larger datasets provide higher accuracy in the results. We applied existing grammar rules and designer expertise to create a few example linear functional chains. These results confirm the notion that the findings of our AFCT algorithm can be used to build a linear functional chain of individual components within a product.

In the second manuscript, we hypothesized that we could improve the data fidelity of the results from our AFCT algorithm by creating a metric that accounts for the prevalence and consistency of the data. In the previous manuscript, we found that CFF frequency was a suitable metric to determine a components likely function and flow. However the limitation of our current automation process is that *prevalence*, the measure of the commonness of the component, and *consistency*, the measure of how uniform the CFF combinations are per component, are not considered. Some CFF combinations appear many times, resulting in high levels of confidence in our results. **Housing, electrical cord,** and **screw** were three components in the repository that appeared well over 100 times in the repository. However, many results only occur once or twice in our dataset, resulting in a lower confidence. In this manuscript, we provided the methodology to create a weighted confidence metric that replaces the common approach of removing rare data. This metric allows all data to be included by describing the data fidelity. Since we were unable to find a numerical tool or quantification that returned the synthesis of prevalence and consistency in our dataset, we developed our own metric. In order to create the weighted confidence metric, we took the harmonic mean of two

metrics, *prevalence*, and *consistency*.

We applied this metric to four example linear functional chains:

1. High CFF frequency, high weighted confidence
2. Low CFF frequency, high weighted confidence
3. Low CFF frequency, low weighted confidence
4. High CFF frequency, low weighted confidence.

We concluded that the components with higher weighted confidence metric indicate higher data fidelity than the components with lower weighted confidence.

The following general findings summarize this research:

Finding 1: The results of the more robust validation methods show that *learning from the most possible products will return a higher accuracy than any restricted-size dataset.* The component-specific datasets had lower accuracy when cross-validated against component-specific data than when cross-validated against all consumer products. The Black and Decker dataset is the smallest, containing 12 products, and consistently had the lowest F1 score when used as the training set. The consumer products dataset is the largest, containing 142 products, and consistently had the highest F1 scores.

Finding 2: We suggest that because the F1 score is calculated for an entire testing set, which often contains rare components that might have only one function and flow in the testing set, this may decrease the overall accuracy of function and flow results per component. As is often the case in large datasets, the accuracy of the data input can be a concern. Over the 20 years of the development of the Design Repository, many different contributors have worked on this project. This

turnover has led to some inconsistencies in the data; for example, container and reservoir are often used interchangeably. Another example, the component *screw* is 91% correlated with *couple solid* but there are 17 other results, which could be due to individual input variations. *This noise of the additional rare or mislabeled CFF combinations in the datasets can certainly reduce the accuracy of the results, especially for the larger consumer products dataset.*

Finding 3: While finding 1 suggests that learning from more data returns more accurate results, restricting the dataset based on the component may return more refined results for functionality. For example, the heating element, and reservoir/container component-specific datasets have six CFF combinations for the component *heating element*, the consumer products dataset has 10, and the blade dataset only returned one result. Heating element and reservoir/container have a high overlap in products, such as coffee makers, but blade products are unlikely to contain heating element as a component. *There may be times when a designer desires more refined results and a smaller learning dataset can be used if the products have the component of interest in the learning set.*

Finding 4: In developing the linear functional chains, we demonstrated more simple examples, such as *screw* and *washer*. As complexity increases, grammar rules are necessary to order the function and flow results. Only two existing grammar rules applied to our findings in heating element and blade. *As we expand our work in developing linear functional chains, we will need to expand on the research around grammar rules to create additional rules required to connect flows at the interface of components.* Individual analysis allows for the development of new

rules to handle each situation, but automation is possible based on investigating the interactions between component, function, and flow. While significant future work is required to fully automate the functional modeling of a product, these findings offer a starting point.

Finding 5: Future work will look at finding the threshold specific to each component where adding additional function-flow combinations has a negligible change. We found the 70% threshold worked for the majority of components, but some components fall outside this typical pattern. For example, the component *screen*, has an even split of 25% frequency across four results; the algorithm automatically sorts the first three results it sees, leaving the last result out arbitrarily. In cases like the above example, future work is needed to improve the algorithm in such cases of uniform distribution. We have explored changing the classification threshold to see how that affects the results, and a more thorough method should be investigated. *Optimizing the classification threshold to individual components would increase the accuracy of the AFCT algorithm results and ultimately the automation process.*

Finding 6: The range of distribution of CFF frequency is shifted towards the lower end of the spectrum because the majority of components in the consumer products dataset have multiple function and flow outputs resulting in the division of CFF frequency across all instances. However, the weighted confidence is distributed more evenly across all CFF combinations, indicating that there is a range of data fidelity in the Design Repository. *Ultimately we need both metrics in order to automate the process; the CFF frequency metric returns the most likely func-*

tion and flow results for each component, and then the weighted confidence metric accounts for prevalence and consistency in the data.

Finding 7: *By including the weighted confidence metric, we have eliminated the tendency to discard useful outliers to reduce the noise in analysis such that these outliers can now be included in the concept generation process.* The inclusion of these outliers can provide valuable creative insight to designers. We have provided a simple method that researchers could implement with their own datasets to weight results versus discarding outliers, ultimately increasing the robustness of data analysis.

Finding 8: We have shown that the AFCT algorithm returns can be used to develop linear functional chains. *In order to broaden linear functional chains to the full functional model, work must be done on connecting components to each other, as well as connecting the components through flows.* Future work should investigate the connections between components by looking at sub-assemblies and assemblies of products in the repository, as well as, the connections of incoming and outgoing flows for each function.

Chapter 7: Conclusion

As this research develops, our ultimate goal is to automate the process of entering additional products into design repositories. Streamlining the process of adding new products with automating functional modeling allows not only individual products to be added by users but also the addition of entire repositories. Enabling products to be entered by users will further increase the size and quality of the data in the Design Repository and ultimately increase the accuracy of our automation process. Short term goals include combining the information from an existing Sustainable Design Repository in the OSU Design Repository [14]. Additionally, Soria et al. have been using Design Repository data to develop new relationships, such as incorporating the user, user interactions, and human error [48] [37]. The database structure of the Design repository provides mapping and connections between categories of the product systems, expanding these connections to include sustainability and user-system interactions will bring these important considerations to the early phase of design decisions.

We posit that this work will have broader impacts by providing designers insight on the functionality of components. These CFF correlations can assist designers and students in building functional models during the concept generation phase, enabling designers to design the whole product, rather than taking a parts up approach. Additionally, this work can also be used by educators to help improve

student understanding of product functionality by connecting function-flow with a component.

Vita

Katherine Edmonds earned her Bachelor of Science degree in Biology from Rhodes College, in Memphis, Tennessee. She earned a Master of Science Degree in Ecology from The University of Georgia. While finishing her Master of Science in Mechanical Engineering, she improved her research skills, 3D CAD design skills, and improved her in depth knowledge about design theory. Her interests are in investigating and improving the design process.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1826469. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Bibliography

[1] Rakesh Agrawal, Tomasz Imieliński, Arun Swami, Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*, volume 22, pages 207–216, New York, New York, USA, 1993. ACM Press.

[2] W Beitz and G Pahl. Engineering design: a systematic approach. *MRS BULLETIN*, 71, 1996.

[3] Matt R. Bohm and Robert B. Stone. A Natural Language to Component Term Methodology: Towards a Form Based Concept Generation Tool. In *Volume 2: 29th Computers and Information in Engineering Conference, Parts A and B*, pages 1341–1350. ASME, jan 2009.

[4] Matt R. Bohm and Robert B. Stone. Form Follow Form - Fine Tuning Artificial Intelligence Methods. *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information*, pages 1–10, 2010.

[5] Matt R Bohm, Robert B Stone, and Robert L Nagel. Form follows form: Is a new paradigm needed? In *ASME 2009 International Mechanical Engineering Congress and Exposition*, pages 165–175. American Society of Mechanical Engineers, 2009.

[6] Matt R. Bohm, Robert B. Stone, Timothy W. Simpson, and Elizabeth D. Steva. Introduction of a data schema to support a design repository. *Computer-Aided Design*, 40(7):801–811, jul 2008.

[7] Matt R. Bohm, Jayson P. Vucovich, and Robert B. Stone. An Open Source Application for Archiving Product Design Information. *Volume 2: 27th Computers and Information in Engineering Conference, Parts A and B*, pages 253–263, 2007.

[8] Cari R. Bryant, Daniel A. McAdams, Robert B. Stone, Jeremy Johnson, Venkat Rajagopalan, Tolga Kurtoglu, and Matthew I. Campbell. Creation

of Assembly Models to Support Automated Concept Generation. In *ASME 2005 International Design Engineering Technical Conferences*, pages 259–266, 2008.

[9] Drew Conway and John Myles White. *Machine Learning for Email: Spam Filtering and Priority Inbox*. O'Reilly Media, Inc., 2011.

[10] Pierre Duchesne and Bruno Rémillard. *Statistical modeling and analysis for complex data problems*, volume 1. Springer Science & Business Media, 2005.

[11] Claudia Eckert and Martin Stacey. Sources of inspiration: a language of design. *Design studies*, 21(5):523–538, 2000.

[12] Katherine Edmonds, Alex Mikes, Robert B Stone, and Bryony DuPont. Data mining a design repository to generate linear functional chains: a step toward automating functional modeling. Under Review, 2020.

[13] Wirth F. Ferger. The Nature and Use of the Harmonic Mean. *Journal of the American Statistical Association*, 26(173):36–40, 1931.

[14] Vincenzo Ferrero, Addison Wisthoff, Tony Huynh, Donovan Ross, and Bryony DuPont. A Sustainable Design Repository for Influencing the Eco-Design of New Consumer Products. *Computer-Aided Design*, page Under Review, 2018.

[15] Katherine Fu, Diana Moreno, Maria Yang, and Kristin L. Wood. Bio-Inspired Design: An Overview Investigating Open Questions From the Broader Field of Design-by-Analogy. *Journal of Mechanical Design*, 136(11):111102, 2014.

[16] Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American statistical Association*, 70(350):320–328, 1975.

[17] Julie Hirtz, Rob Stone, Daniel McAdams, Simon Szykman, and Kristin Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13:65–82, 2002.

[18] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

[19] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14, 03 2001.

[20] Mark A. Kurfman, Michael E Stock, Robert B Stone, Jagan Rajan, and Kristin L Wood. Experimental Studies Assessing the Repeatability of a Functional Modeling Derivation Method. *Journal of Mechanical Design*, 125:682–693, 2003.

[21] Tolga Kurtoglu and M. I. Campbell. Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping. *Journal of Engineering Design*, 20(1):83–104, feb 2009.

[22] Tolga Kurtoglu, Matthew I Campbell, Cari R Bryant, Robert B Stone, and Daniel A McAdams. Deriving a Component Basis for Computational Functional Synthesis. *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy*, page 4061, 2005.

[23] Tolga Kurtoglu, Matthew I Campbell, Cari R Bryant, Robert B Stone, Daniel A McAdams, et al. Deriving a component basis for computational functional synthesis. In *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy*, page 1687. Engineers Australia, 2005.

[24] Pier L Lanzi. *Learning classifier systems: from foundations to applications*. Number 1813. Springer Science & Business Media, 2000.

[25] Shu Hsien Liao and Chih H. Wen. Mining demand chain knowledge for new product development and marketing. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 2009.

[26] Alberto Lora-Michiels, Camille Salinesi, and Raúl Mazo. A Method based on Association Rules to Construct Product Line Model. *4th International Workshop on Variability Modelling of Software-intensive Systems (VaMos)*, page 50, jan 2010.

[27] Alex Mikes, DuPont Bryony Edmonds, Katherine, and Robert B Stone. Optimizing and algorithm for data mining a design repository to automate functional modeling. Under Review, 2020.

[28] Scarlett R. Miller and Brian P. Bailey. Searching for Inspiration: An In-Depth Look at Designers Example Finding Practices. In *Volume 7: 2nd Biennial International Conference on Dynamics for Design; 26th International Conference on Design Theory and Methodology*, page V007T07A035. ASME, aug 2014.

[29] Robert L. Nagel, Jayson P. Vucovich, Robert B. Stone, and Daniel A. McAdams. A Signal Grammar to Guide Functional Modeling of Electromechanical Products. *Journal of Mechanical Design*, 130(5):051101, may 2008.

[30] Salvatore Orlando, P. Palmerini, and Raffaele Perego. Enhancing the Apriori algorithm for frequent set counting. *Data Warehousing and Knowledge Discovery.*, 2114:71–82, 2001.

[31] KN Otto and Kristin Wood. *Product design: techniques in reverse engineering and new product development*. 2003.

[32] Bryan M OHalloran, Chris Hoyle, Robert B Stone, and Irem Y Tumer. The early design reliability prediction method. In *ASME 2012 International Mechanical Engineering Congress and Exposition*, pages 1765–1776. American Society of Mechanical Engineers Digital Collection, 2012.

[33] Bryan M OHalloran, Chris Hoyle, Robert B Stone, and Irem Y Tumer. A method to calculate function and component failure distributions using a hierarchical bayesian model and frequency weighting. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 727–736. American Society of Mechanical Engineers Digital Collection, 2012.

[34] Kerstin Sailer, Ros Pomeroy, and Rosie Haslem. Data-driven design-Using data on human behaviour and spatial configuration to inform better workplace design. Technical Report 3.

[35] NIST Sematech. Engineering statistics handbook. *NIST SEMATECH*, 2006.

[36] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer, 2006.

[37] Nicolás F. Soria Zurita, Robert B. Stone, H. Onan Demirel, and Irem Y. Tumer. Identification of HumanSystem Interaction Errors During Early Design Stages Using a Functional Basis Framework. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 6(1), mar 2020.

[38] Prasanna Sridharan and Matthew I Campbell. A study on the grammatical construction of function structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 19(3):139–160, 2005.

[39] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

[40] Robert B. Stone and Kristin L. Wood. Development of a Functional Basis for Design. *Journal of Mechanical Design*, 122(4):359, dec 2000.

[41] Ke Sun and Fengshan Bai. Mining weighted association rules without preassigned weights. *IEEE transactions on knowledge and data engineering*, 20(4):489–495, 2008.

[42] Daniel Svensson and Johan Malmqvist. Integration of Requirement Management and Product Data Management Systems. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 1, pages 199–208, 2001.

[43] S. Szykman, R.D. Sriram, C. Bochenek, J.W. Racz, and J. Senfaute. Design repositories: engineering design's new knowledge base. *IEEE Intelligent Systems*, 15(3):48–55, may 2000.

[44] Melissa Tensa, Katherine Edmonds, Vincenzo Ferrero, Alex Mikes, Nicolas Soria Zurita, Rob Stone, and Bryony DuPont. Toward Automated Functional Modeling: An Association Rules Approach for Mining the Relationship between Product Components and Function. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1):1713–1722, jul 2019.

[45] David G Ullman. *The mechanical design process*, volume 2. McGraw-Hill New York, 1992.

[46] Wei Wang, Jiong Yang, and Philip S Yu. Efficient mining of weighted association rules (war). In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 270–274, 2000.

[47] Maria C Yang. Observations on concept generation and sketching in engineering design. *Research in Engineering Design*, 20(1):1–11, 2009.

[48] Nicolás F Soria Zurita, Robert B Stone, Onan Demirel, and Irem Y Tumer. The function-human error design method (fhedm). In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V007T06A058–V007T06A058. American Society of Mechanical Engineers, 2018.

APPENDICES

Appendix A: Functional Model Example

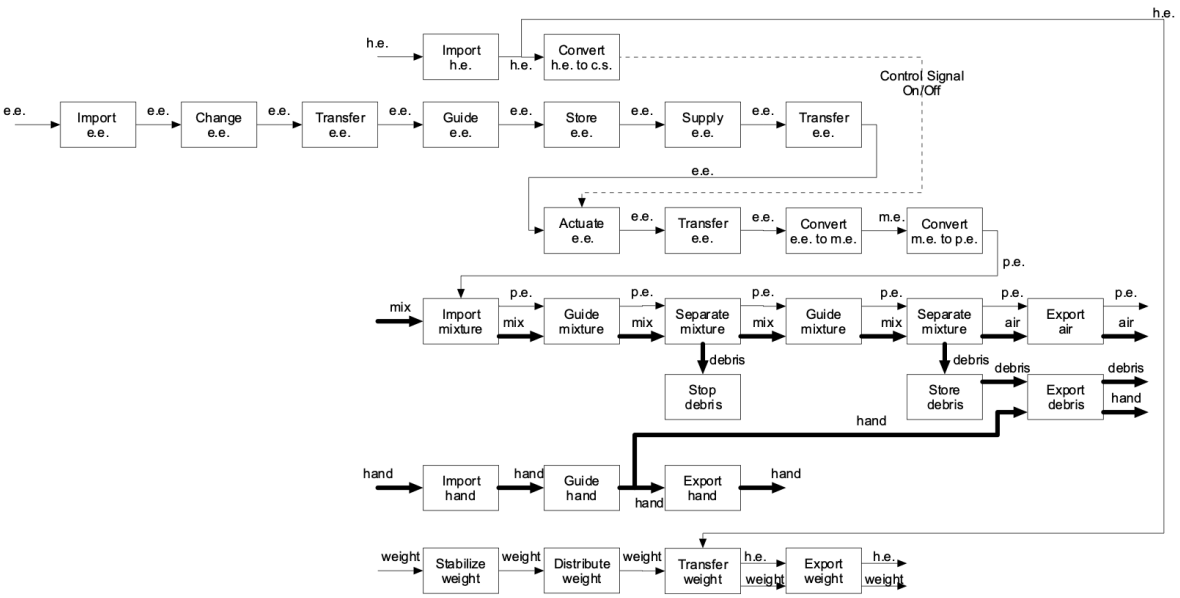


Figure A.1: Black And Decker Dustbuster Functional Model

Appendix B: Additional Component Examples

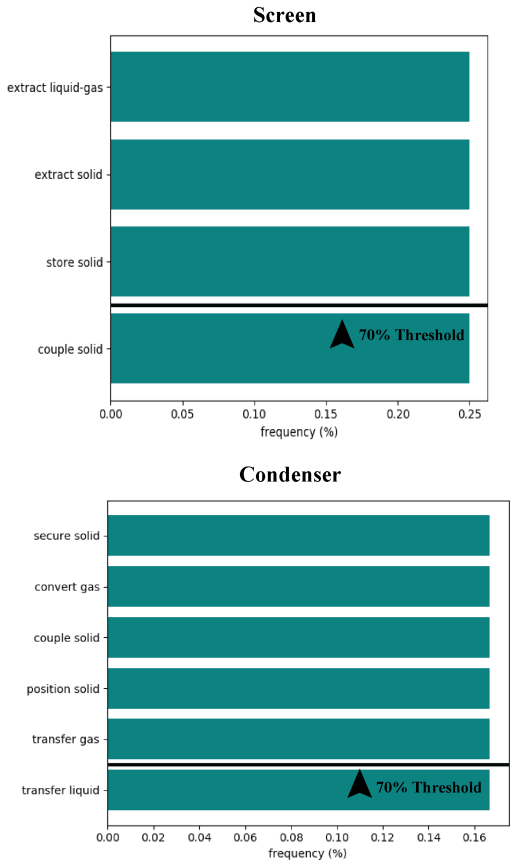


Figure B.1: Example Components To Illustrate Limitations Of Threshold Automation