Proceedings of the ASME 2020 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2020
August 16-19, 2020, St. Louis, USA

IDETC2020-22495

A WEIGHTED CONFIDENCE METRIC TO IMPROVE AUTOMATED FUNCTIONAL MODELING

Katherine Edmonds Alex Mikes, Bryony DuPont, Robert B. Stone*
Design Engineering Laboratory
School of Mechanical, Industrial and Manufacturing Engineering
Oregon State University
Corvallis, Oregon, 97331
Email: Edmondka@oregonstate.edu
Email: MikesA@oregonstate.edu
Email: Bryony.DuPont@oregonstate.edu
Email: Rob.Stone@oregonstate.edu*

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.

1 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 [1]. 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

*Corresponding author.

[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

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 [2]. 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 [3,4]. Consistency and uniformity are built into the process with the widely accepted use of the Functional Basis and Component Basis terms [5–7]. However, even with consistency built into the process, functional models can vary widely by the individual user input [8]. 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 [9]. However, research has shown that concept generation is more robust when the function is considered [3]. 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 [10]. 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 [11]. 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.

2 BACKGROUND

2.1 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 [1, 13–16]. Each product is divided into seven main categories of design information: artifact, function, failure, physical, performance, sensory, and media-related information types [17]. A visual reference of the data schema (i.e., the connections between data) is shown in Figure 1. These different levels of abstraction to help improve design knowledge and data-driven design decisions [1]. We define data-driven design as methodologies for extracting information and insights from data and existing research to improve design processes [18].

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 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 [15]. The function and flow utilize the Functional Basis terms developed by Stone and Hirtz [5, 6]. 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.

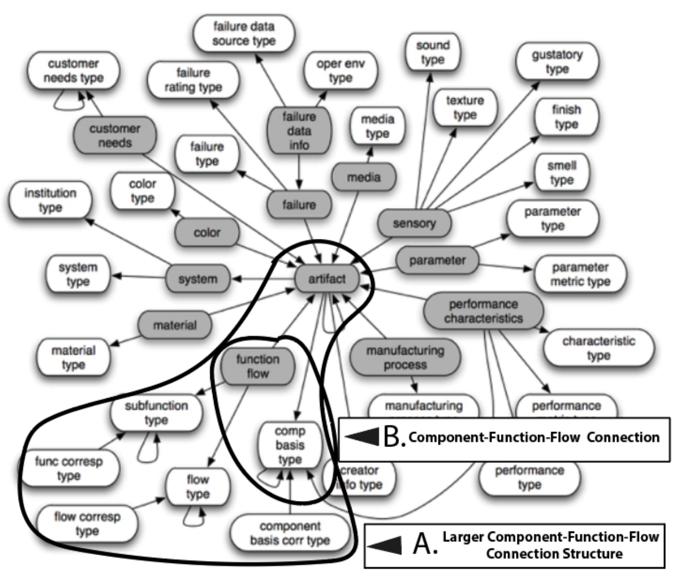

FIGURE 1: Design Repository Data Schema [17]

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 [10, 19]. In the FFF approach, Bohm et al. calculated the CFF frequency of function and flow associated with components *separately*. Our research continues to build on this concept, attempting to streamline the automation process by *combining* the component-function-flow association referred 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. Our method is more thorough and complete investigation of the data than previous work due to the addition of the new weighted confidence metric which allows us to determine the prevalence and consistency of the data.

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 [11]. Association rules are a type of data mining that describes the relationship between items in item sets [20] [21]. 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 [19]. 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 7 in the Appendix. Figure 7 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 [11, 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 [3, 4]. Some of this research involves developing grammar rules to help solve the consistency issue with building functional models [8, 16]. Kurfman et al. found that despite a formal language, repeatability was a challenge among both novices and experts [22]. 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.

2.2 Weighted Confidence Metric

Weighting is a commonly used tool when dealing with statistical probabilities or uncertainty [23, 24]. 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.

O'Halloran et al. developed a frequency weighting metric that helps understand reliability and uncertainty in early design phases [25]. 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) [26, 27]. 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 [28].

Our method of calculating a weighted confidence metric is similar to the EDRPM because it accounts for occurrence (prevalence) and is similar to Inverse Probability Weighting (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 [29].

3 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.

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 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 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

TABLE 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

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 automation 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*.

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

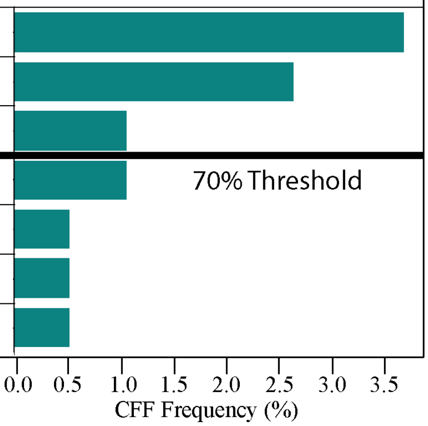

FIGURE 2: EXAMPLE TO ILLUSTRATE THRESHOLD AUTOMATION FOR THE COMPONENT PULLEY

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 1. Two example components demonstrate the numbers used to calculate the weighted confidence metric and are shown in Table 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 1 is a more representative mean when dealing with ratios rather than the arithmetic mean [29], 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.

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

High prevalence is demonstrated in Table 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 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 2. As

TABLE 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**

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 2, provides additional information improving our automation process. High CFF frequency indicates that the component has few functions and flows associated 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 [19]. 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 8 in the Appendix shows the AFCT algorithm results for both components. In cases like this example and the *pulley* example (Figure 2), future work is needed to optimize the threshold in the AFCT algorithm.

4 RESULTS AND DISCUSSION

4.1 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 3.

4.2 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 3 trend line.

Figure 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 4 shows that the weighted confidence metric is needed to

TABLE 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

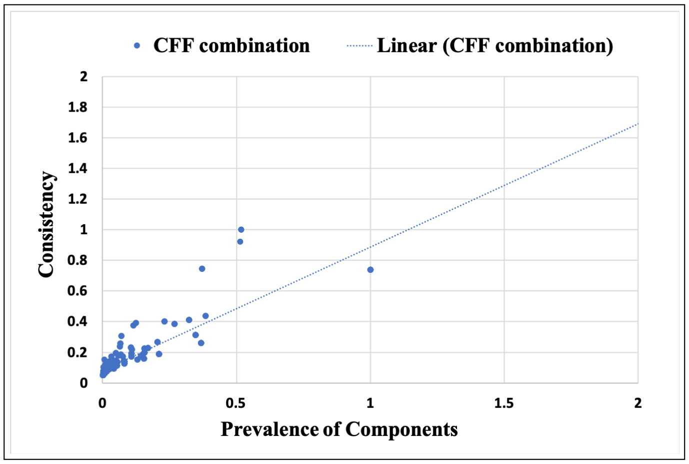

FIGURE 3: CONSISTENCY VERSUS PREVALENCE

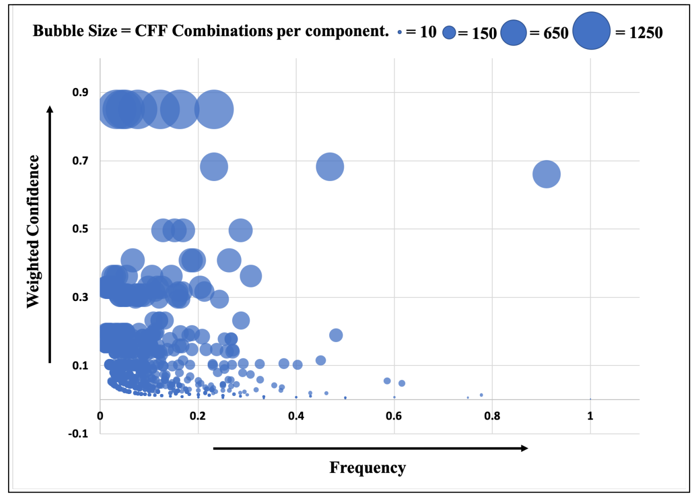

FIGURE 4: WEIGHTED CONFIDENCE VERSUS CFF FREQUENCY WITH THE OCCURRENCE OF THE COMPONENT AS THE SIZE OF THE BUBBLE

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 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, 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 2. The results demonstrate 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.

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 combinations 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.

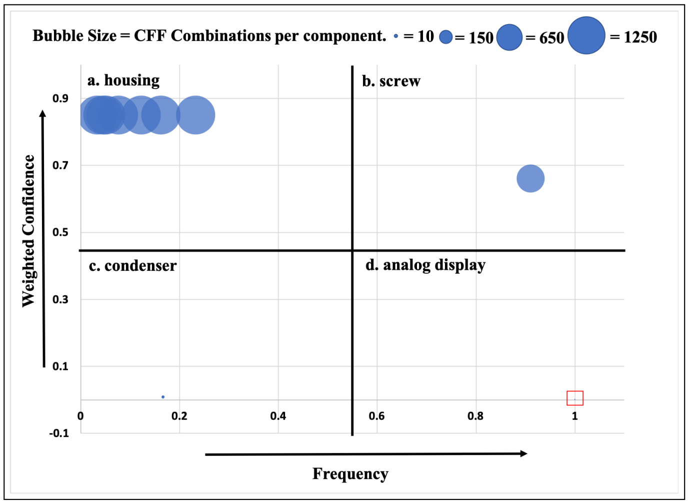

FIGURE 5: EXAMPLE DATA FOR THE FOUR QUADRANTS OF COMBINED WEIGHTED CONFIDENCE AND CFF FREQUENCY.

4.3 Linear Functional Models

To translate our findings into automation, we developed four linear functional chains based on our four examples above in Figure 5. Each example component came from one of the four quadrants shown in Table 2. The linear functional 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 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 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 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 [19]. 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, Figure6 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 6.

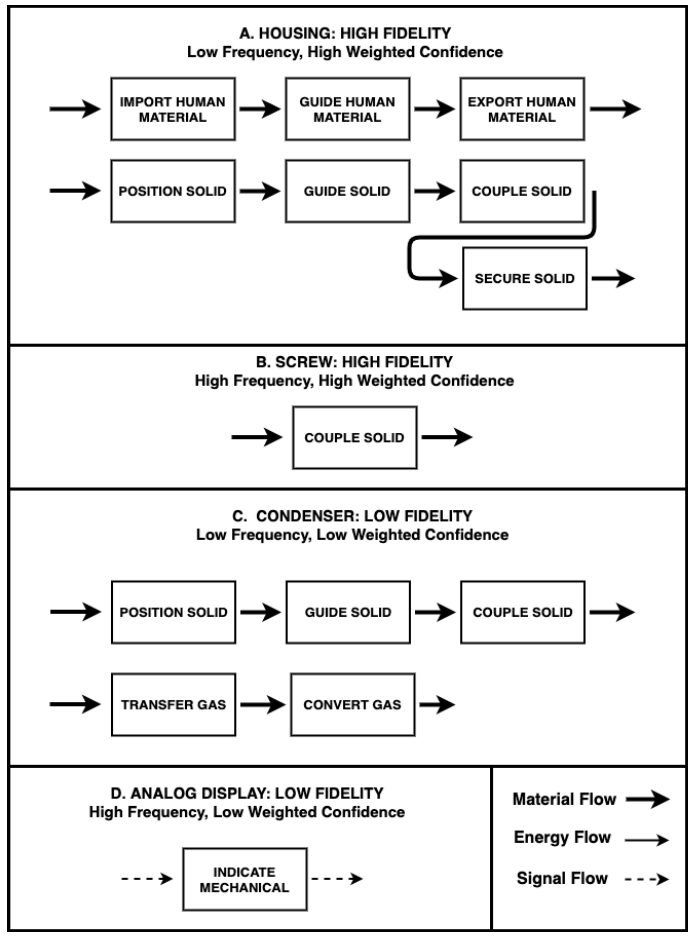

FIGURE 6: LINEAR FUNCTIONAL CHAINS OF THE FOUR EXAMPLES IN FIGURE 5

5 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 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 modeling. 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 Oregon State University (OSU) researchers to house the information from an existing Sustainable Design Repository in the OSU Design Repository [30]. Combining this information adds additional products, as well as sustainable design information such as Life Cycle Analysis (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 [31, 32]. 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.

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.

REFERENCES

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

[2] Yang, M. C., 2009. "Observations on concept generation and sketching in engineering design". *Research in Engineering Design,* ***20***(1), pp. 1–11.

[3] Ullman, D. G., 1992. *The mechanical design process*, Vol. 2. McGraw-Hill New York.

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

[5] Stone, R. B., and Wood, K. L., 2000. "Development of a Functional Basis for Design". *Journal of Mechanical Design,* ***122***(4), dec, p. 359.

[6] Hirtz, J., Stone, R., McAdams, D., Szykman, S., and Wood, K., 2002. "A functional basis for engineering design: Reconciling and evolving previous efforts". *Research in Engineering Design,* ***13***, pp. 65–82.

[7] Kurtoglu, T., Campbell, M. I., Bryant, C. R., Stone, R. B., McAdams, D. A., et al., 2005. "Deriving a component basis for computational functional synthesis". In ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy, Engineers Australia, p. 1687.

[8] Nagel, R. L., Vucovich, J. P., Stone, R. B., and McAdams, D. A., 2008. "A Signal Grammar to Guide Functional Modeling of Electromechanical Products". *Journal of Mechanical Design,* ***130***(5), may, p. 051101.

[9] Miller, S. R., and Bailey, B. P., 2014. "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, ASME, p. V007T07A035.
[10] Bohm, M. R., Stone, R. B., and Nagel, R. L., 2009. "Form follows form: Is a new paradigm needed?". In ASME 2009 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers, pp. 165–175.
[11] Tensa, M., Edmonds, K., Ferrero, V., Mikes, A., Soria Zurita, N., Stone, R., and DuPont, B., 2019. "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), jul, pp. 1713–1722.
[12] Edmonds, K., Mikes, A., Stone, R. B., and DuPont, B., 2020. "Data mining a design repository to generate linear functional chains: a step toward automating functional modeling". In DCC 2020 Ninth Interational Conference on Design Computing and Cognition.
[13] Bohm, M. R., Vucovich, J. P., and Stone, R. B., 2007. "An Open Source Application for Archiving Product Design Information". *Volume 2: 27th Computers and Information in Engineering Conference, Parts A and B*, pp. 253–263.
[14] Bryant, C. R., McAdams, D. A., Stone, R. B., Johnson, J., Rajagopalan, V., Kurtoglu, T., and Campbell, M. I., 2008. "Creation of Assembly Models to Support Automated Concept Generation". In ASME 2005 International Design Engineering Technical Conferences, pp. 259–266.
[15] Kurtoglu, T., Campbell, M. I., Bryant, C. R., Stone, R. B., and McAdams, D. A., 2005. "Deriving a Component Basis for Computational Functional Synthesis". *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy*, p. 4061.
[16] Sridharan, P., and Campbell, M. I., 2005. "A study on the grammatical construction of function structures". *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM,* ***19***(3), pp. 139–160.
[17] Bohm, M. R., Stone, R. B., Simpson, T. W., and Steva, E. D., 2008. "Introduction of a data schema to support a design repository". *Computer-Aided Design,* ***40***(7), jul, pp. 801–811.
[18] Sailer, K., Pomeroy, R., and Haslem, R. Data-driven design-Using data on human behaviour and spatial configuration to inform better workplace design. Tech. Rep. 3.
[19] Bohm, M. R., and Stone, R. B., 2010. "Form Follow Form - Fine Tuning Artificial Intelligence Methods". *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information*, pp. 1–10.
[20] Orlando, S., Palmerini, P., and Perego, R., 2001. "Enhancing the Apriori algorithm for frequent set counting". *Data Warehousing and Knowledge Discovery.,* ***2114***, pp. 71–82.
[21] Liao, S. H., and Wen, C. H., 2009. "Mining demand chain knowledge for new product development and marketing". *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*.
[22] Kurfman, M. A., Stock, M. E., Stone, R. B., Rajan, J., and Wood, K. L., 2003. "Experimental Studies Assessing the Repeatability of a Functional Modeling Derivation Method". *Journal of Mechanical Design,* ***125***, pp. 682–693.
[23] Wang, W., Yang, J., and Yu, P. S., 2000. "Efficient mining of weighted association rules (war)". In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 270–274.
[24] Sun, K., and Bai, F., 2008. "Mining weighted association rules without preassigned weights". *IEEE transactions on knowledge and data engineering,* ***20***(4), pp. 489–495.
[25] O'Halloran, B. M., Hoyle, C., Stone, R. B., and Tumer, I. Y., 2012. "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, American Society of Mechanical Engineers Digital Collection, pp. 727–736.
[26] Holt, C. C., 2004. "Forecasting seasonals and trends by exponentially weighted moving averages". *International journal of forecasting,* ***20***(1), pp. 5–10.
[27] Sematech, N., 2006. "Engineering statistics handbook". *NIST SEMATECH*.
[28] O'Halloran, B. M., Hoyle, C., Stone, R. B., and Tumer, I. Y., 2012. "The early design reliability prediction method". In ASME 2012 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers Digital Collection, pp. 1765–1776.
[29] Ferger, W. F., 1931. "The Nature and Use of the Harmonic Mean". *Journal of the American Statistical Association,* ***26***(173), pp. 36–40.
[30] Ferrero, V., Wisthoff, A., Huynh, T., Ross, D., and DuPont, B., 2018. "A Sustainable Design Repository for Influencing the Eco-Design of New Consumer Products". *Computer-Aided Design*, p. Under Review.
[31] Zurita, N. F. S., Stone, R. B., Demirel, O., and Tumer, I. Y., 2018. "The function-human error design method (fhedm)". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V007T06A058–V007T06A058.
[32] Soria Zurita, N. F., Stone, R. B., Onan Demirel, H., and Tumer, I. Y., 2020. "Identification of Human–System 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.

A Appendix A: Functional Model Example

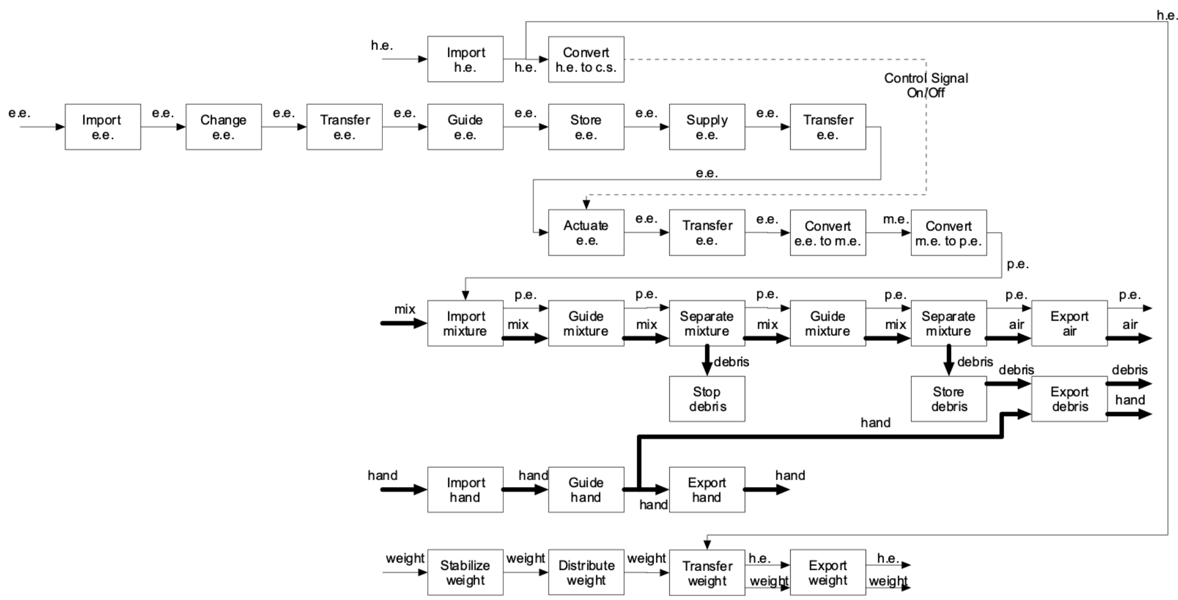

FIGURE 7: BLACK AND DECKER DUSTBUSTER FUNCTIONAL MODEL

B Appendix B: Additional Component Examples

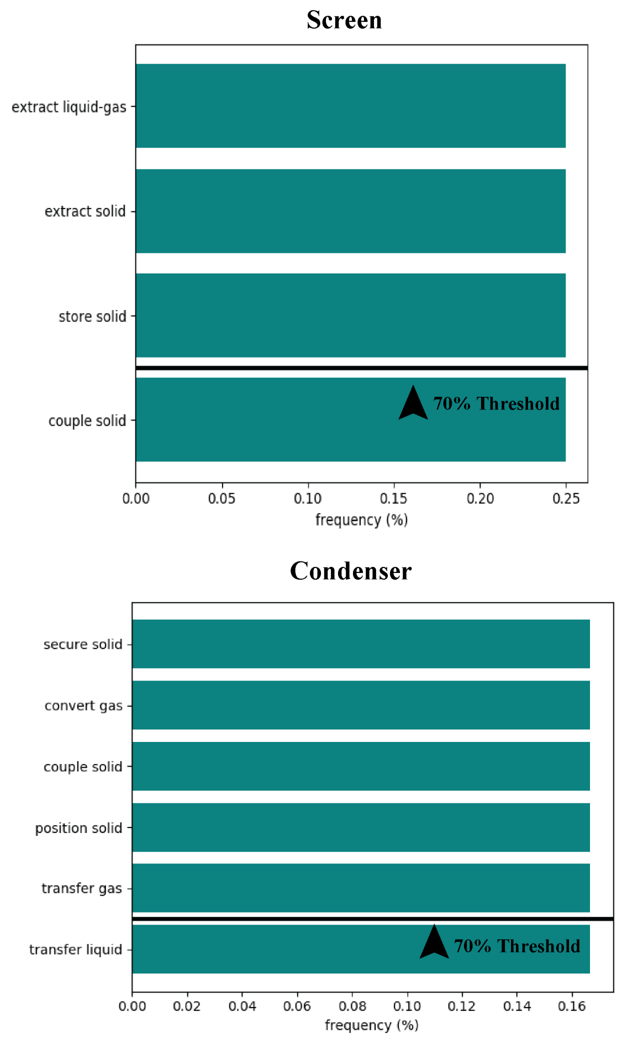

FIGURE 8: EXAMPLE COMPONENTS TO ILLUSTRATE LIMITATIONS OF THRESHOLD AUTOMATION