A Computational Technique for Concept Generation

Article · January 2005

DOI: 10.1115/DETC2005-85323

CITATIONS
90

READS
842

5 authors, including:

Daniel A. Mcadams
Texas A&M University
171 PUBLICATIONS **3,704** CITATIONS
SEE PROFILE

Tolga Kurtoglu
Palo Alto Research Center
67 PUBLICATIONS **1,767** CITATIONS
SEE PROFILE

Matthew I. Campbell
Oregon State University
163 PUBLICATIONS **2,318** CITATIONS
SEE PROFILE

Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-85323

A COMPUTATIONAL TECHNIQUE FOR CONCEPT GENERATION

Cari R. Bryant
Department of Mechanical & Aerospace Engineering
University of Missouri – Rolla
Rolla, Missouri, 65409, USA
crb5ea@umr.edu

Daniel A. McAdams
Department of Mechanical & Aerospace Engineering
University of Missouri – Rolla
Rolla, Missouri, 65409, USA
dmcadams@umr.edu

Robert B. Stone
Department of Basic Engineering
University of Missouri – Rolla
Rolla, Missouri, 65409, USA
rstone@umr.edu

Tolga Kurtoglu
Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas, 78712, USA
tolga@mail.utexas.edu

Matthew I. Campbell
Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas, 78712, USA
mc1@mail.utexas.edu

ABSTRACT

Few computational tools exist to assist designers during the conceptual phase of design, and design success is often heavily weighted on personal experience and innate ability. Many well-known methods (e.g. brainstorming, intrinsic and extrinsic searches, and morphological analysis) are designed to stimulate a designer's creativity, but ultimately still rely heavily on individual bias and experience. Under the premise that quality designs comes from experienced designers, experience in the form of design knowledge is extracted from existing products and stored for reuse in a web-based repository. This paper presents an automated concept generation tool that utilizes the repository of existing design knowledge to generate and evaluate conceptual design variants. This tool is intended to augment traditional conceptual design phase activities and produce numerous feasible concepts early in the design process.

Keywords: concept generation, design tools, Functional Basis, functional models

1. INTRODUCTION

The concept generation phase of the design process is, at best, difficult to translate into a succinct methodology that is useful to both experienced and inexperienced designers. Quantification and formalization of the conceptual design phase is an active, but immature, area of research. Many formal methods of conceptual design have yet to be realized as computational algorithms. The work presented in this paper shows the initial steps taken to generate an automated, mathematically based algorithm for concept generation and early concept evaluation. The specific focus of this research is the combination and formalization of function-based synthesis, constraint management, and state space search to create a comprehensive space of concept variants and search it for feasible design candidates.

Existing design tools primarily focus on the initial design phases, such as customer need gathering (e.g. quality function deployment), or on the later steps of design embodiment or detail design (e.g. design structure matrices, graph grammars, solid models, dynamic modeling, and finite element analysis.) Few computational tools exist to assist designers during the conceptual phase of design. Instead, designers have limited options available for creating a quality design. Available options may include drawing on personal experiences or the experiences of co-workers, utilizing patent searches to find other approaches or similar designs, and reverse engineering existing products to evaluate how either the current design or a redesign could be used to meet the design goals. All of these methods are potentially limited or biased by a designer's experiences. In addition, patent searches and reverse engineering are often time intensive and laborious and may not catch solutions that seem unrelated but are, in fact, analogous.

Innovations cited by Antonnsson and Cagan [1] indicate that certain parts of larger design problems can be solved automatically and without human expertise. However, automations in the design

process are often only employed once basic design concepts have been selected but lack specific dimensions. Complete automation of the design process seems to be restricted by a lack of continuity between conceptual design methods and computational design tools. We propose a computationally based method of concept generation that quickly produces a manageable array of concept variants. The automatically generated concept variants can then be used for concept selection or as a catalyst for generating additional concept variants through complimentary non-computational creative techniques. The following sections present a review of the product representation and design tools that have been used in this paper. We then present our computer-implemented algorithm for automatically producing conceptual designs and illustrate its effective use to generate viable concept variants.

2. REVIEW OF RELATED WORK

We begin with a review of the state of the art in area of conceptual design research and areas that support automated concept generation. In particular, we first review systematic approaches to conceptual design and then focus on product function representation and design knowledge collection.

The fuzzy front end of the conceptual design process has seen few attempts at automation, perhaps due to the evolving strategies and methodologies that exist for this phase of design. However, over the past few decades, design methods have matured and systematic approaches to conceptual design have emerged [2-4]. These design methods provide a starting point for automating the conceptual design phase. In particular, the systematic approach of Pahl and Beitz and Hubka [5], representing European schools of design, has spawned variant methodologies in American design literature [3, 6-12]. Regardless of the methodology variation, all begin by formulating the overall product function and breaking it into small, easily solved sub-functions. Solutions to the sub-functions are sought and the form of the device then follows from the assembly of all sub-function solutions.

The lack of a precise definition for *small, easily solved sub-functions* has spurred research into the development of a high level design language (sometimes called a vocabulary or taxonomy) to describe product function and thus enable a systematic approach to functional modeling. Hundal [13] formulates six function classes complete with more specific functions in each class, though he does not claim to have an exhaustive list of mechanical design functions. Another approach uses the 20 subsystem representations from living systems theory to represent mechanical design functions [14]. Malmqvist, *et al*. [15] compare the Soviet Union era design methodology known as the Theory of Inventive Problem Solving (TIPS) with the Pahl and Beitz methodology. TIPS uses a set of 30 functional descriptions to describe all mechanical design functions [16]. Malmqvist, *et al*. note that the detailed vocabulary of TIPS would benefit from a more carefully structured class hierarchy using the Pahl and Beitz functions at the highest level. Kirschman and Fadel [17] propose four basic mechanical functions groups, but vary from the standard verb-object sub-function description common to most methodologies. This work appears to be the first attempt at creating a common vocabulary of design that leads to common functional models of products. Japanese researchers have also explored a consistent language for describing the functionality of products and relating it to product behavior [18-21].

Recent work continues this pursuit for a standard language that subsumes the previous work [12, 22-26]. The result of these recent efforts is a design language known as the Functional Basis. The Functional Basis uses the function and flow words in Tables 1 & 2 to form a sub-function description as a *function* and a *flow* (i.e., a verb-object format). The Functional Basis is intended to be broad enough to span the entire mechanical design space while not being repetitive. In Table 1, engineering *flows* are categorized as three classes (material, signal and energy) and then further specified as basic categories within each class. In Table 2, engineering *functions* are categorized as 8 classes that are further specified as basic categories.

Table 1: *Flow classes and their basic categorizations.*

Class	Material	Signal	Energy		
Secondary	Human	Status	Human	Electrical	Mechanical
	Gas	Signal	Acoustic	Electromagnetic	Pneumatic
	Liquid		Biological	Hydraulic	Radioactive
	Solid		Chemical	Magnetic	Thermal
	Plasma				
	Mixture				

Table 2: *Function classes and their basic categorizations.*

Class	Branch	Channel	Connect	Control Magnitude	Convert	Provision	Signal	Support
Secondary	Separate	Import	Couple	Actuate	Convert	Store	Sense	Stabilize
	Distribute	Export	Mix	Regulate		Supply	Indicate	Secure
		Transfer		Change			Process	Position
		Guide		Stop				

A more formal approach to functional representation stores customer needs and functions in a matrix format [27, 28]. This process is shown schematically in Figure 1. In this work, products are represented quantitatively as vectors that indicate the functionality present in each product. Multiple products are aggregated together to form a product-function matrix, Φ. Matrix manipulations of Φ produce useful design knowledge such as similarity of products based on functionality and customer needs (using the formulation $\Lambda=\Phi^T\Phi$). This computational approach provides a way to 'design by analogy,' i.e., use stored product knowledge to design new or redesign existing products.

Building on this representation, a prototypical design repository following the NIST-proposed format has been developed [25, 29, 30]. The repository, in its current form, holds information on approximately 50 consumer products and, following the NIST schema, identifies artifact, function, flow, form, geometry, material, behavior and specification for each product in an XML format [31].

From a perspective different than the functional modeling ap-

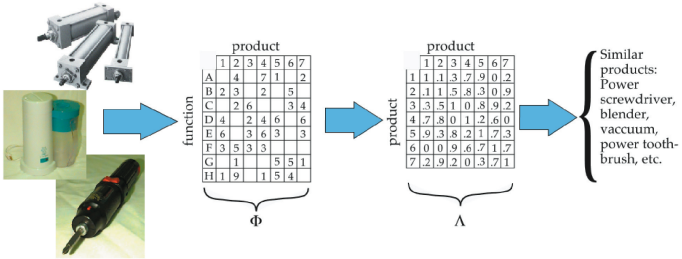

Figure 1: *A schematic representation of creating a product-function repository and its manipulation to produce a product-product matrix and the corresponding product families.*

proach discussed above, a number of research efforts have sought to establish a generic computational scheme for electromechanical design. While these methods have yet to capture *function* on the same level understood by human designers, such approaches have been used in attempts to synthesize new electromechanical configurations. These methods use a variety of computer techniques including case-based reasoning [32], constraint programming [33], qualitative symbolic algebra [34] or geometric algebras [35]. One of the most historically significant of these includes several approaches applying expert system formulations to specific design problems such as the paper roller system established by Mittal, *et al*. [36]. In the approaches reviewed, the repeating refrain is that computational synthesis approaches produce an overwhelming number of concept variants – even for limited domains – and a pruning method is needed to realistically identify all feasible solutions [37, 38].

The concept generation program illustrated in Section 4 utilizes computational strategies [39] comparable to the above matrix techniques [27, 28] to reuse existing stored design knowledge to quickly produce and evaluate many feasible concepts.

3. REVIEW OF DESIGN TOOLS USED IN THE CONCEPT GENERATION SOFTWARE

The following sections describe the design tools used to automate the concept generation phase of the design process. First, the Functional Basis is presented as the means of both capturing the design knowledge and representing the conceptual design input required by the computational scheme. Next, the web-based design repository used to store the design knowledge is described. Lastly, the matrix manipulations that comprise the computational concept generation algorithm are described.

3-1. FUNCTIONAL BASIS OF DESIGN

Intended to span the entire mechanical design space without repetition, the Functional Basis uses *function* and *flow* words to form a sub-function description as a function and a flow (i.e., a verb-object format). Generation of a black box model, creation of function chains for each input flow, and aggregation of function chains into a functional model are the sequence of steps that lead to the repeatable formation of a functional model in their approach [40, 41]. To briefly illustrate this technique, the functional model of an insulating cup is shown in Figure 2. The black box model is

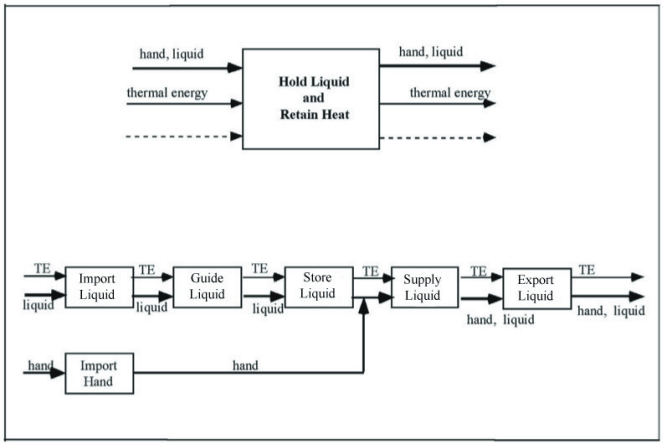

Figure 2: *Black box model and functional model of a cup.*

constructed based on the overall product function and includes the various energy, material, and signal flows involved in the global functioning of the product. The detailed functional model is then derived from sub-functions that operate on the flows listed in the black box model.

Functional models for any product can be generated using this technique. Repeatability, ease in storing and sharing design information, and increased scope in the search for solutions are some of the advantages of these functional models [42]. Functional models reveal functional and flow dependencies and are used to capture design knowledge from existing products for inclusion in the web-based design repository [31]. Functional representations also increase the clarity of the design problem and tracking of input and output flows [2].

3-2. DESIGN REPOSITORY

Over the course of several years, a web-based repository to store design knowledge has been developed and refined at the University of Missouri–Rolla and in collaboration with the University of Texas at Austin [43]. This repository, which includes descriptive product information such as functionality, bills of materials, and design structure matrices (DSMs), now contains detailed design knowledge on approximately 50 consumer products. The knowledge contained in the repository is steadily expanding and benefits from a broad base of consumer products. Design tools like function component matrices (FCMs) and design structure matrices (DSMs) can be readily generated from single or multiple products and used in a variety of ways to enhance the design process [31, 43]. These design tools have recently been enhanced by incorporating a recently revamped *component naming basis* to classify product artifacts [44]. Each artifact is classified under a specific component name according to its function. For example, separate artifacts under different products may be named "motor 1", "shaded pole induction motor", or "dc motor". Using the updated component basis naming scheme, each of the of these artifacts would be identified as similar and tagged as an "electric motor". Using this basis allows for well-defined function-based groupings of artifacts to be used in the creation of FCMs and DSMs, helping to maintain matrices of manageable size and eliminating artifact redundancies that may not be immediately evident due to variations in user-dependent artifact naming. By eliminating these redundancies, a larger variety of unique and more abstract concept variants can be quickly generated and evaluated using these design tools. After concept variants are selected using the generalized component basis names, individual artifacts classified classified under the chosen component basis names can be inspected to spur more specific concept variant ideas. For example, if a returned concept variant included an "electric motor", the repository could be accessed to provide the designer with the specific examples "motor 1", "shaded pole induction motor", or "dc motor". Next, we discuss more specifically how these design repository tools can be utilized to produce viable concept variants.

3-3. CONCEPT GENERATION ALGORITHM

Our proposed method utilizes the Functional Basis to link component functionality with component compatibility and create,

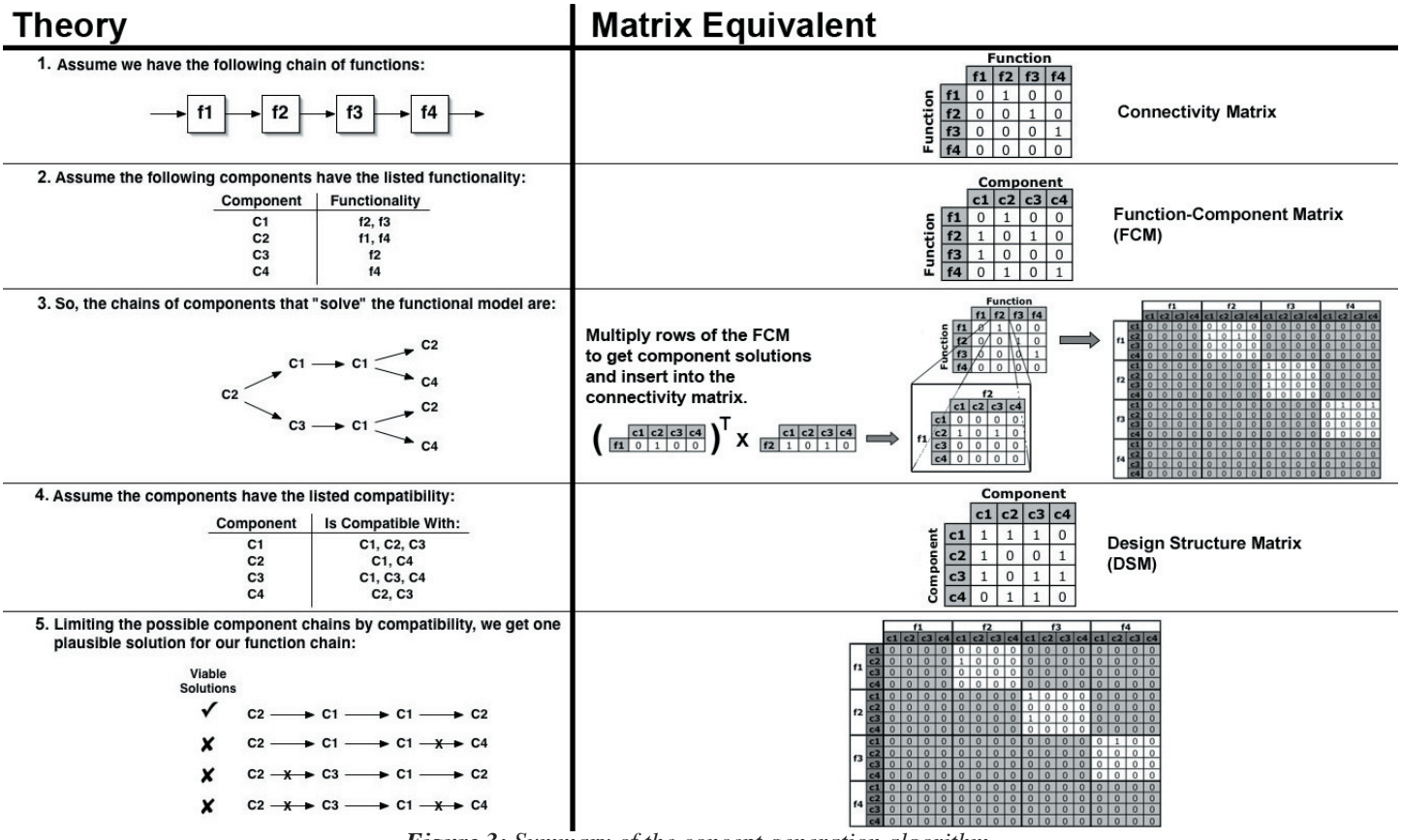

Figure 3: *Summary of the concept generation algorithm.*

filter, and rank concept variants [26, 45]. Generated from a web-based repository of design information, the function-component matrix (FCM) and the design structure matrix (DSM) describe the function-component relationships and the component-component compatibility of existing consumer products [8, 45]. Product descriptions stored in the database allow access to information such as historical occurrence and failure mode, which help limit and rank design solutions. Because of the potential for an automated method to generate more concept variants than is reasonable for individual analyses, detailed knowledge such as this is necessary to help a designer identify the most promising results. Functions comprising a proposed product's functional model are mapped to lists of components that are capable of solving each function. The tree of possible component chains is then pruned by eliminating infeasible component connections according to component-component compatibility [39]. Figure 3 briefly illustrates the philosophy behind the design tool and correlates the Functional Basis matrices used to generate and filter the conceptual design variants to the theory behind the concept generation algorithm.

Step 1 under Theory in Figure 3 shows a simple generic flow chain of the form used to create functional models using the Functional Basis method, where f1-f4 are unspecified sub-functions of the product to be designed. The matrix equivalent of this flow chain is a connectivity matrix where a non-zero cell entry indicates a forward connection between the row and column functions. Step 2 under Theory in Figure 3 shows the component/function relationship encompassed in the function-component matrix generated from the design repository. In step 3 under Theory in Figure 3, a component "tree" is created showing the chains of components that could potentially solve the flow chain presented in step 1, based on the information contained in the FCM shown in step 2. Although the example illustrated in Figure 3 results in a single branching tree for step 2, it is important to note that multiple branching trees may be formed at this stage when multiple components have the potential to solve the initiating function in the chain. In this case, this method executes an exhaustive search in order to capture all potential solutions from each of the branching chains. Computationally, if the transpose of the row vector from the FCM that corresponds to each of the functions from the flow chain in step 1 is matrix multiplied by the row vector from the FCM that corresponds to the forward connected function, a component-component matrix will be generated for each function connection in the flow chain. Non-zero cells within these newly created component-component matrices represent all theoretically possible component combinations that will solve each pair of functions in the flow chain. If these component-component matrices are then placed into the connectivity matrix, component paths can be traced through the aggregated matrix much the same way a path is traced along the tree shown in step 3 under Theory in Figure 3. Step 4 under Theory shows the component/component relationship encapsulated by the design structure matrix generated from the design repository. In step 5 under Theory, each cell of the DSM is multiplied with the corresponding cell in each of the function pair component-component matrices generated in step 3. Overlaying the DSM on each matrix created in step 3 (via cell multiplication) has the effect of removing any of the possible component connections that do not appear in the repository database. This technique uses the "experience" contained in the repository

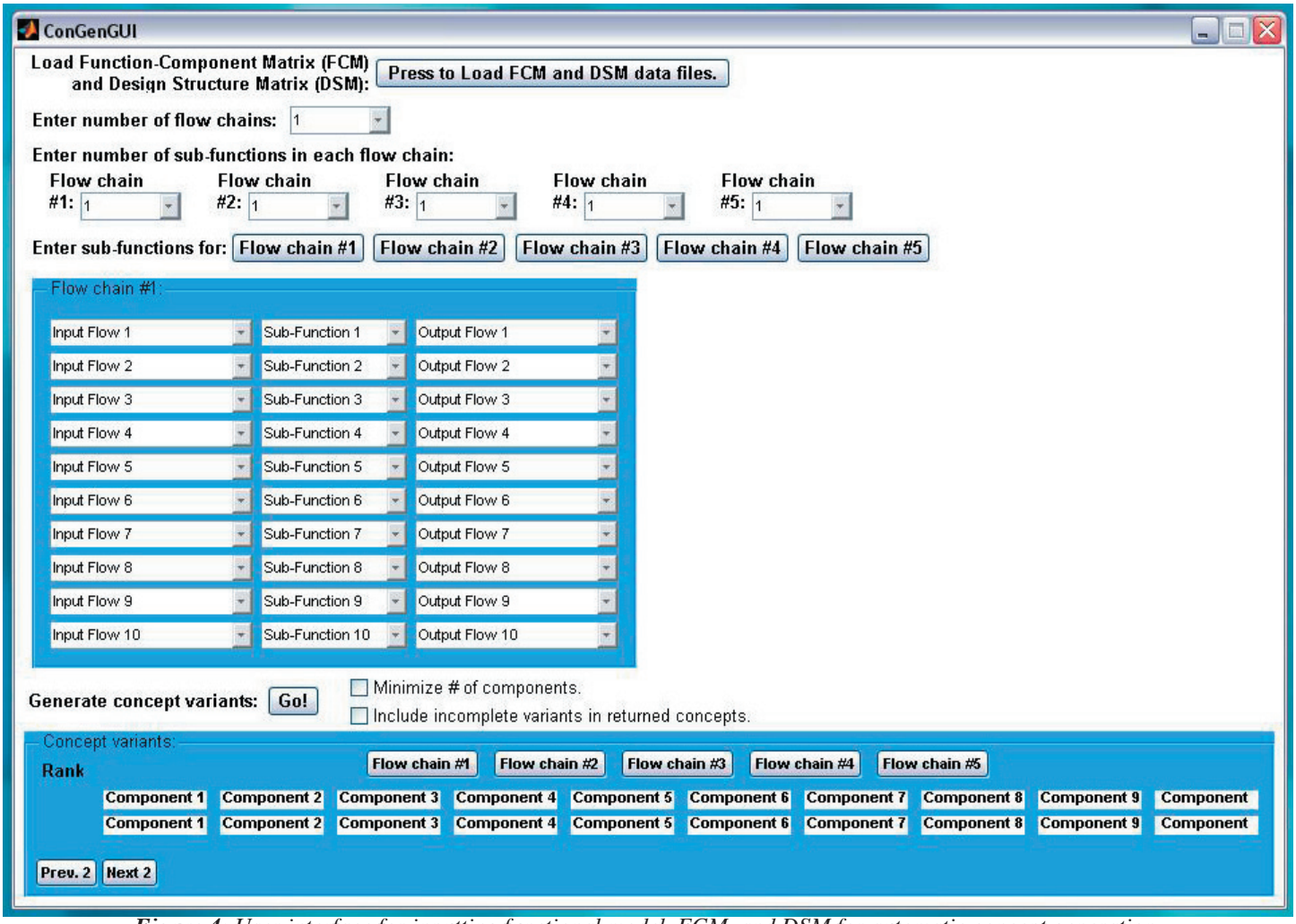

Figure 4: *User interface for inputting functional model, FCM, and DSM for automatic concept generation.*

to filter out potentially inadequate concept variants and reduce the set of possible concept variants down to a more manageable size. Furthermore, various measures of design needs (e.g. manufacturability, recyclability, failure etc.) entered as the non-zero FCM and/or DSM entries can be used to rank the resulting conceptual design solutions generated by this method. Once the set of filtered concept variants has been computed and ranked, a designer is then free to sift through the generated concept variants and evaluate the application of each to the design situation at hand.

4. AUTOMATION OF THE CONCEPT GENERATION ALGORITHM

Using the algorithm described in Section 3-3, a Matlab [46] program was created to automatically produce a ranked list of concept variants for an input functional model chain. The user interface, shown in Figure 4, firsts prompts the user for the location of the tab-delimited function-component matrix (FCM) and design structure matrix (DSM) files generated from the design repository of existing products from which the new concepts will be created. The FCM and DSM design tools in the repository permit the user to select any subset of products from the repository from which to generate these matrices.

Next, the user enters the number of distinct flow chains contained in the conceptual functional model. This initial version of the concept generation software limits flow chain entries to five separate non-branching flows. The user then selects the number of sub-functions in each flow chain and proceeds to enter the input and output flows and sub-functions for each individual chain. At this point, concepts can be generated and ranked for each flow chain by selecting the "Go!" button. The number of components displayed for each concept variant can be minimized by selecting the "Minimize # of components" checkbox. Under the pretense that a single component has the potential to solve multiple functions, selecting this option instructs the program to search for repeating series of components in the concept variant chain and collapse them down to a single instance for display. The option to include incomplete variants in the ranked returned concepts is also available. This allows the user to decide whether to display concept variant chains that may be incomplete (i.e. not all sub-functions have an associated component solution) since the design repository may not yet contain preexisting solutions for the entered flow/sub-function combination. If selected, incomplete variants will show a question mark in chains where a solution cannot be found. Once the concept variants are created and ranked, the user can browse through them using the interface at the bottom of the screen.

After obtaining the user input, the program automatically filters the FCM to contain only those functions relevant to the

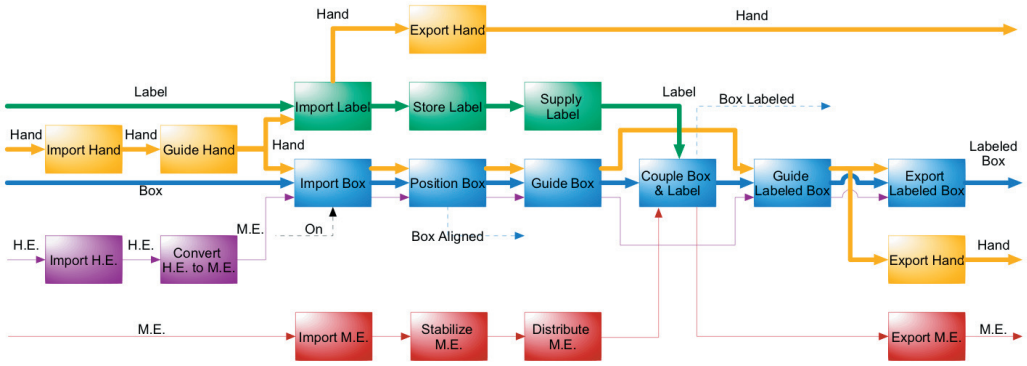

Figure 5: *(Above) Conceptual functional model for the case study of a box-labeling device.*

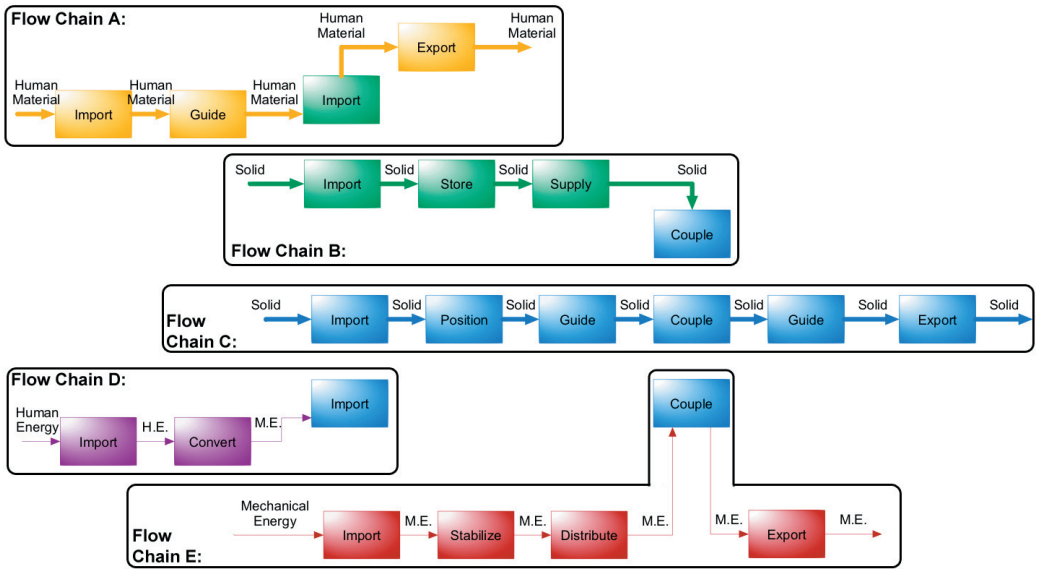

Figure 6: *The conceptual functional model was divided into single non-branching flow chains, labeled Flow Chain A-E, and entered into the program.*

user-input functional model. From this filtered FCM, or morphological matrix, the component-component matrices for each pairing of functions are calculated and filtered using the information contained in the DSM. Finally, all combinations of the remaining feasible component-component connections are determined, ranked, and output as potential component configurations for the input functional model. By using them as a point of departure for other non-computational creative techniques like brainstorming, these conceptual design variants can then be further developed and/or modified by the designer to satisfy the design requirements. The next section presents a case study for the creation of a box-labeling device to demonstrate the effectiveness of the software in a real-world design situation.

5. CASE STUDY

This section presents a case study that demonstrates how the automated concept generation software can effectively assist a designer during the early phases of design. A design team was charged with creating a box-labeling device to assist workers at a local area workshop for persons with disabilities. Prior to the designer's solution, the task of labeling the contents of cardboard boxes filled with sample products from a local business was restricted to those workers who possessed the agility and mental capacity required to properly handwrite the information on the box. The managers at the workshop were looking for a solution that would allow any of the workers to perform this task regardless of level of ability, but while maintaining a level of quality acceptable to the local business who contracted the work. After determining the applicable customer needs for the device to be designed, the conceptual functional model, shown in Figure 5, for the box-labeling device was generated.

Since the current form of the software is limited to handling single, non-branching flow chains, the functional model, shown in Figure 5, was divided into the color coded chains, as illustrated in Figure 6. Note that sub-functions with multiple input/output flows appear in multiple flow chains, and that these repeated sub-functions appear immediately vertically adjacent to each other in Figure 6. These five flows chains were used as the input into the concept generation program.

A function component matrix (FCM) and a design structure matrix (DSM) were generated from the design knowledge of 50 consumer products contained in the web-based repository. These tab-delimited data files were then loaded into the program to seed the concept generation scheme. As described in Section 3-3, data from the FCM was used to generate the concept variants, while data from the DSM was used to filter the set of vaiants to those

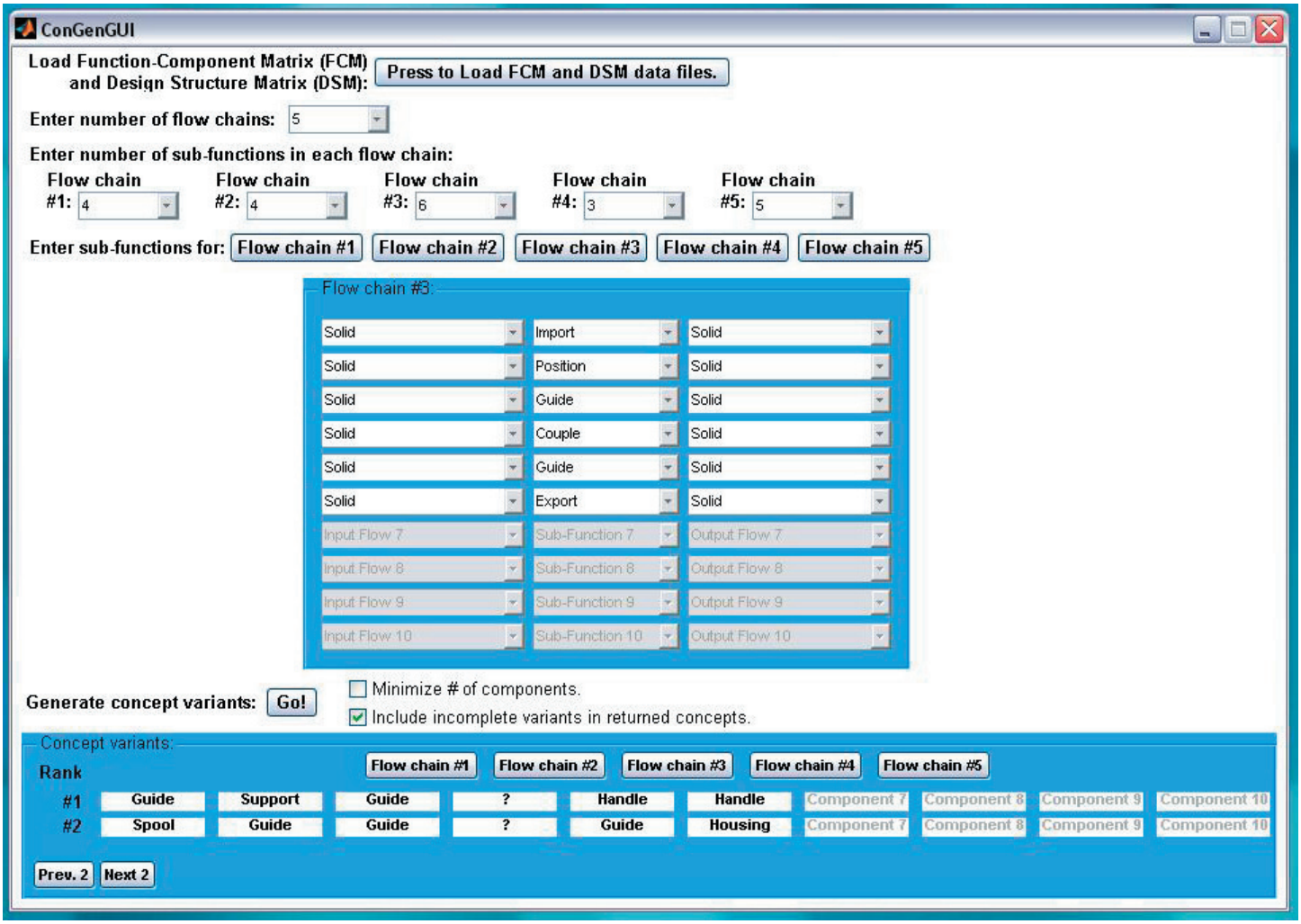

Figure 7: *(Above) Completed user interface form for the function chains shown in Figure 6. Buttons located above the input flow, sub-function, and output flow entries allow the user to navigate between displays for the various flow chains.*

with known viability. The magnitude of the cell values from the FCM were used to rank the concepts by decreasing value of historical occurrance. Next, the data from the user-generated function model, shown in Figure 6, was input into the program GUI. Finally, concepts for each chain were generated. The completed user interface form for flow chain C (from Figure 6) is shown in Figure 7.

The display at the bottom of the form shown in Figure 7 demonstrates how the top two ranked concepts are displayed for flow chain C. Additional concept variants for flow chain C can be viewed using the "Prev. 2" "Next 2" advance buttons located at the bottom of the section labeled "Concept variants:". Taken from the information displayed at the bottom of the user interface form, the first and second ranked component chains for each flow chain, A-E, were gathered and displayed in Figure 8. Components outlined in the vertical red boxes indicate components solving sub-functions that are present in more than one chain. For clarity, Table 3 on the followiing page shows the definitions for the subset of component basis names shown in Figure 8. Program functionality to be included in later versions will include the ability to browse through specific examples of each component basis named artifact in the resulting concept variant chains. For instance, clicking on

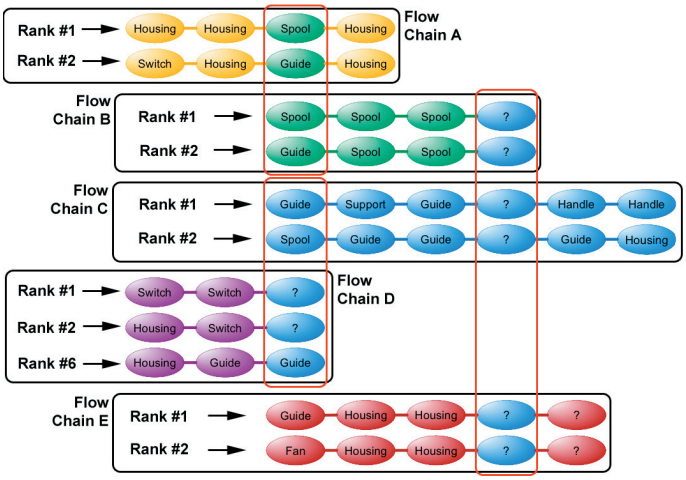

Figure 8: *Conceptual component chains generated from the program. Components grouped together vertically by the red boxes indicate overlap in the component chains. This redundancy is triggered when the complete functional model is divided into individual flow chains, causing singular sub-functions to appear in multiple flow chains.*

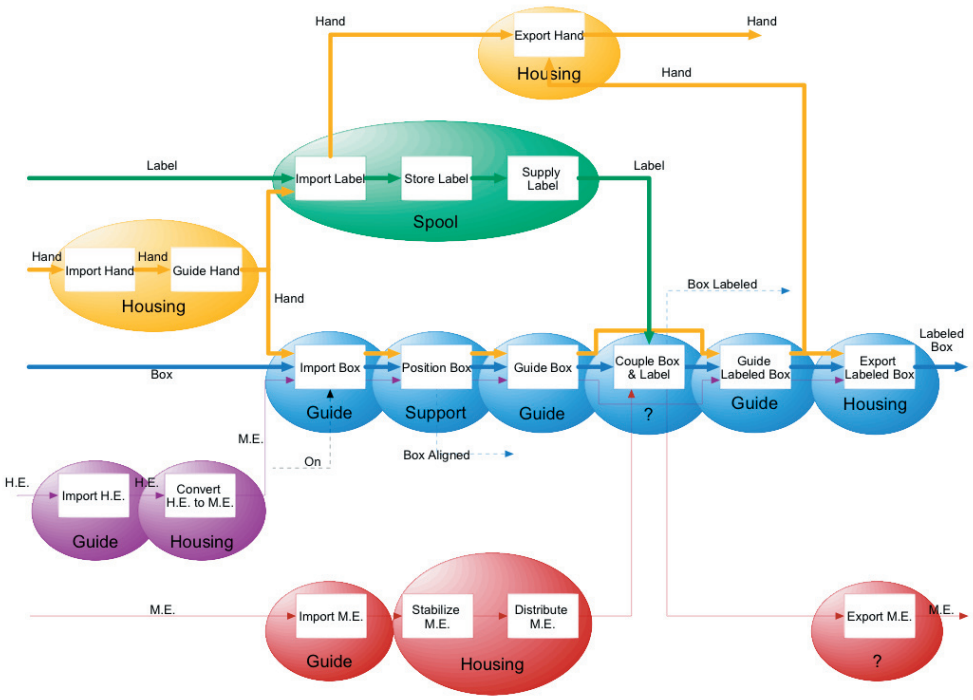

Figure 9: *(Above) Aggregated concept variant generated from the component chains shown in Figure 8. Components are associated with the sub-functions from the functional model they solve.*

the returned component "guide" may produce a list of example artifacts tagged as "guides" under the component naming basis, such as a "column", "rail", or "threading fixture". The user would then be able to browse through these examples to gather ideas or more detailed information for a specific concept variant.

The individual component chains, shown in Figure 8, can then be reassembled to produce a complete concept variant for the product to be redesigned (see Figure 9). To help clarify the component-function relationships for the concept variant chosen, the complete concept variant, shown in Figure 9, was overlaid onto the function model from Figure 5. Sketching techniques can next be employed as a final step to create a drawing of the selected conceptual design variant. Using the component basis naming descriptions and pictures from the web-based repository of specific artifacts as guides, specific embodiments of the conceptual design idea were generated for the box-labeling device by sketching various configurations of the returned component basis artifact names, one of which is shown in Figure 10.

Many concepts were generated using various methodologies during the course of the box-labeling project. After all of the concept variants generated from the various methods were evaluated and ranked, the sketch shown in Figure 10 was used as the starting point for the final box-labeling device design. Although the eventual embodiment of the box-labeling device, shown in Figure 11, was modified during the later stages of design from the initial conceptual sketch presented in Figure 10, the concept variant shown catalyzed the idea that led to a successful end design.

Table 3: *Subset of component basis artifacts found in the listed concept variants [44].*

Component Basis Name	Definition
Fan	A device composed of blades around a revolving hub.
Guide	Any device by which another object is led in its proper course.
Handle	A component that allows any action that is thought of as comparable to grasping something or keeping it in place.
Housing	A device fitted to contain or enclose other devices or items.
Spool*	
Support	Anything that holds up, or sustains the weight of a body. (includes beam, excludes bracket)
Switch	Control consisting of a mechanical or electrical or electronic device for making or breaking or changing the connections in a circuit.

** specific artifact label; not tagged under component basis*

6. CONCLUSIONS AND FUTURE WORK

This paper automates a mathematically based concept generation technique [39] developed from an empirical study of consumer products [45]. Intended to help designers choose a correct component for a given function in a redesign or original design situation, the goal is to utilize existing design knowledge to rapidly produce a large array of concepts early in the design process. Compared to traditional concept generation methods, the process presented here is quick and does not require the effort of an entire team of designers. Future improvements to the software design include adding the capability to generate concepts for a full (branching) functional model. Achieving this goal will include coding the algorithm using a more flexible software platform that supports the design of an enhanced user interface and allows for a more seamless integration of the application into the existing web-based repository. Furthermore, the implemented algorithms need to be carefully evaluated for efficiency given the potential need to quickly manipulate, search, and sort large matrices to be useful. As briefly mentioned in previous sections, additional features of future software versions should include the exploration of various ranking methods to help sort the concept variants generated. Although using the design structure matrix as a filter eliminates many less useful concepts from the set of design variants, metrics such as measures of failure, manufacturing and assembly costs, quality, recyclability, or some mathematical combination of similar design characteristics could prove to be invaluable tools for identifying the most promising variants among the hundreds (or thousands)

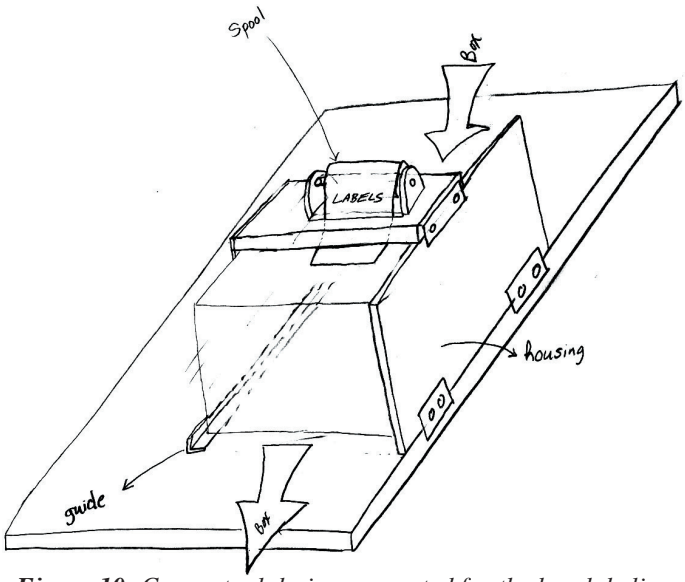

Figure 10: *Conceptual design generated for the box-labeling device, inspired by the concept generation program output.*

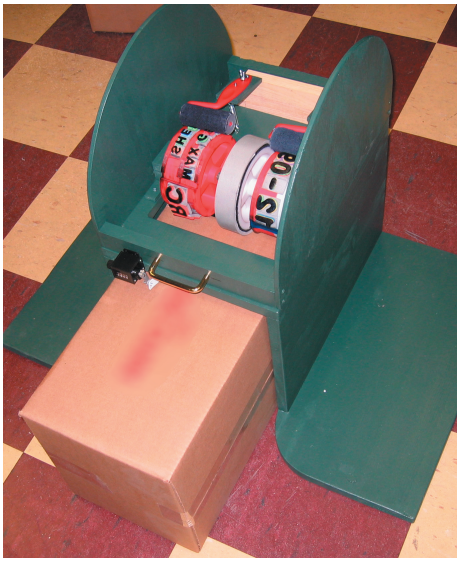

Figure 11: *Embodied design for the box-labeling device.*

of viable solutions found. Further areas of expansion include mapping functional models to assemblies (groups of components) in addition to individual components. Also, the program should be capable of expanding beyond a one-to-one mapping of components to functions and of capturing component chains that may require an intermediate component for connection. Although basic in its current form, the concept generation program presented has great potential for development into a powerful design tool.

7. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grant NSF IIS-0307419. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators.

8. REFERENCES

1. Antonsson, E. K., Cagan, J., 2001, Formal Engineering Design Synthesis, Cambridge University Press.

2. Pahl, G. and Beitz, W., 1988, Engineering Design: A Systematic Approach, Springer-Verlag.

3. Ulrich, K. and Eppinger, S., 1995, Product Design and Development, McGraw-Hill.

4. Otto, K. and Wood, K., 2001, Product Design: Techniques in Reverse Engineering and New Product Development, Prentice-Hall.

5. Hubka, V. and Ernst Eder, W., 1984, Theory of Technical Systems, Springer-Verlag, Berlin.

6. Ullman, D., 1997, The Mechanical Design Process 2nd ed., McGraw-Hill.

7. Schmidt, L. and Cagan, J., 1995, "Recursive Annealing: A Computational Model for Machine Design," Research in Engineering Design, 7(2):102-125.

8. Pimmler, T. and Eppinger, S., 1994, "Integration Analysis of Product Decompositions," Proceedings of the ASME Design Theory and Methodology Conference, DE-Vol. 68.

9. Shimomura, Y., Tanigawa, S., Takeda, H., Umeda, Y., Tomiyama, T., 1996, "Functional Evaluation Based on Function Content," Proceedings of the 1996 ASME Design Theory and Methodology Conference, 96-DETC/DTM-1532, Irvine, CA.

10. Cutherell, D., 1996, "Chapter 16: Product Architecture," The PDMA Handbook of New Product Development, M. Rosenau Jr. *et al.*, ed., John Wiley and Sons.

11. Otto, K. and Wood, K., 1996, "A Reverse Engineering and Redesign Methodology for Product Evolution," Proceedings of the 1996 ASME Design Theory and Methodology Conference, 96-DETC/DTM-1523, Irvine, CA.

12. Otto, K. and Wood, K., 1997, "Conceptual and Configuration Design of Products and Assemblies," ASM Handbook, Materials Selection and Design, Vol. 20, ASM International.

13. Hundal, M., 1990, "A Systematic Method for Developing Function Structures, Solutions and Concept Variants," Mechanism and Machine Theory, 25(3):243-256.

14. Koch, P., Peplinski, J., Allen, J. and Mistree, F., 1994, "A Method for Design Using Available Assets: Identifying a Feasible System Configuration," Behavioral Science, 30:229-250.

15. Malmqvist, J., Axelsson, R., and Johansson, M., 1996, "A Comparative Analysis of the Theory of Inventive Problem Solving and the Systematic Approach of Pahl and Beitz," Proceedings of the 1996 ASME Design Engineering Technical Conferences, 96-DETC/DTM-1529, Irvine, CA.

16. Altshuller, G., 1984, Creativity as an Exact Science, Gordon and Branch Publishers.

17. Kirschman, C. and Fadel, G., 1998, "Classifying Functions for Mechanical Design," Journal of Mechanical Design, Transactions of the ASME, 120(3):475-482.

18. Kitamura, Y. and Mizoguchi, R., 1998, "Functional Ontology for Functional Understanding," Twelfth International Workshop on Qualitative Reasoning (QR-98), AAAI Press, pp. 77-87, Cape Cod, Massachusetts.

19. Kitamura, Y. and Mizoguchi, R., 1999, "Metafunctions of Artifacts," Proceedings of the Thirteenth International Workshop on Qualitative Reasoning (QR-99), pp. 136-145, Loch Awe, Scotland.

20. Umeda, Y. and Tomiyama, T., 1997, "Functional Reasoning in Design," IEEE Expert, March-April, pp. 42-48.

21. Sasajima, M., Kitamura, Y., Ikeda, M. and Mizoguchi, R., 1995, "FBRL: A Function and Behavior Representation Language," Proceedings of IJCAI'95, pp. 1830-1836.

22. Little, A., Wood, K., and McAdams, D., 1997, "Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking and Redesign," Proceedings of the 1997 Design Engineering Technical Conferences, 97-DETC/DTM-3879, Sacramento, CA.

23. Stone, R. and Wood, K., 1999, "Development of a Functional Basis for Design," Proceedings of DETC99, DETC99/DTM-8765, Las Vegas, NV.

24. Murdock, J., Szykman, S. and Sriram, R., 1997, "An Information Modeling Framework to Support Design Databases and Repositories," Proceedings of DETC'97, DETC97/DFM-4373, Sacramento, CA.

25. Szykman, S., Racz, J., and Sriram, R., 1999, "The Representation of Function in Computer-Based Design," Proceedings of DETC99, DETC99/DTM-8742, Las Vegas, NV.

26. 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(2):65-82.

27. McAdams, D., Stone, R., and Wood, K., 1999, "Functional Interdependance and Product Similarity based on Customer Needs," Research in Engineering Design, 11(1)1-19.

28. Stone, R., Wood, K., and Crawford, R., 2000, "Using Quantitative Functional Models to Develop Product Architectures," Design Studies 21(3):239-260.

29. Szykman, S., 2002, "Architecture and Implementation of a Design Repository System," Proceedings of DETC2002, DETC2002/CIE-34463, Montreal, Canada.

30. NIST Workshop on Product Representation for Next-Generation Distributed Product Development, National Institute of Standards and Technology, Gaithersburg, MD, Nov. 30 - Dec. 1, 2000.

31. Bohm, M. and Stone, R., 2003, "Refining Design Repositories: Creating a Usable Framework with XML Data Representation," Proceedings of the 2003 NSF Grantees Conference, Birmingham, AL.

32. Navinchandra, D., Sycara, K. P., and Narasimhan, S., 1991, "A Transformational Approach to Case-Based Synthesis", AI EDAM, Vol. 5, pp. 31-45.

33. Subramanian, D. and Cheuk-San Wang, 1995, "Kinematic Synthesis with Configuration Spaces," Research in Engineering Design, Vol.7, no.3, p. 193-213.

34. Williams, B.C., 1990, "Interaction-based invention: designing novel devices from first principles", AAAI-90 Proceedings. Eighth National Conference on Artificial Intelligence, Vol.1, Boston, MA, pp. 349-356.

35. Palmer, R. S., and Shapiro, V., 1993, "Chain Models of Physical Behavior for Engineering Analysis and Design," Research in Engineering Design, Vol. 5, pp. 161-184.

36. Mittal, S., Dym, C., and Morjara, M., 1985, "PRIDE: An Expert system for the Design of Paper Handling Systems", IEEE Computer, Vol.19, No.7, pp. 102-114.

37. Chakrabarti, A. and Bligh, T., 1996, "An Approach to Functional Synthesis of Mechanical Design Concepts: Theory, Applications and Emerging Research Issues," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10:313-331.

38. Umeda, Y., Ishii, M., Yoshioka, M., Shiomura, Y. and Tomiyama, T., 1996, "Supporting Conceptual Design Based on the Function-Behavior-State Modeler," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10:275-288.

39. Bryant, C., McAdams, D., Stone, R., Campbell, M., Kurtoglu, T., 2005, "Concept Generation from the Functional Basis of Design", Accepted at ICED'05, Melbourne, Australia.

40. McAdams, D., Stone, R., and Wood, K., 1998, "Understanding Product Similarity Using Customer Needs," Proceedings of DETC98, DETC98/DTM-5660, Atlanta, GA.

41. Kurfman, M., Rajan, J., Stone, R. and Wood, K., 2001 "Functional Modeling Experimental Studies," Proceedings of DETC2001, DETC2001/DTM-21709, Pittsburgh, PA.

42. Stone, R. and Wood, K., 2000, Development of a Functional Basis for Design, Journal of Mechanical Design, 122(4):359-370.

43. Bohm, M., and Stone, R., 2004, "Representing Functionality to Support Reuse: Conceptual and Supporting Functions," Proceedings of DETC'04, DETC2004-57693, Salt Lake City, UT.

44. Kurtoglu, T., Campbell, M., Bryant, C., Stone, R., 2005, "Deriving a Component Basis for Computational Functional Synthesis", Accepted at ICED'05, Melbourne, Australia.

45. Strawbridge, B., McAdams, D. and Stone, R., 2002, "A Computational Approach To Conceptual Design," Proceedings of DETC2002, DETC2002/DTM-34001, Montreal, Canada.

46. Matlab, © The Mathworks, Inc. 1994-2005. www.mathworks.com.