An experimental study on the effects of a computational design tool on concept generation

Tolga Kurtoglu and *Matthew I. Campbell,* Automated Design Laboratory, Department of Mechanical Engineering, One University Station C2200, University of Texas at Austin, Austin, Texas 78712-0292, USA
Julie S. Linsey, Innovation, Design Reasoning, Engineering Education & Methods Lab, Department of Mechanical Engineering, Texas A&M University, 3123 TAMU, College Station, TX 77843, USA

We have developed a computational design tool to help designers create conceptual solutions to detailed functional specifications. The computational method extracts design knowledge from an expanding online design library in the form of procedural rules, and provides these rules as the building blocks for solving new problems. In this paper, we study how this automated approach would benefit designers during concept generation. Accordingly, we test the effects of using our computational tool as an aid for concept generation in an experiment mimicking real design scenarios. Three metrics (completeness, novelty and variety) are used to evaluate the solutions generated to two separate design problems in order to determine how effective the computational method outputs are in improving conceptual design generation.

Keywords: conceptual design, creative design, design knowledge, graph grammars

Concept generation in real engineering design problems is a very human process in which a designer or a team of designers work together to develop both practical as well as unconventional concepts. In recent years, industries have strived to establish a casual and free environment for such concept generation (Kelly and Littman, 2001). The motivation for such environments has been to foster the creative spirits of the designers or rather to provide them avenues to play and explore far-reaching conceptual solutions to both new design problems (that have no well accepted solution) and re-design problems (in which past solutions may be simply refined to meet the design goals). The value of a productive concept generation phase is often evident in the quality of the final product, and industry is well motivated to provide designers with the utmost freedom at this stage of the design process.

Corresponding author:
Matthew I. Campbell
mc1@mail.utexas.edu

www.elsevier.com/locate/destud
0142-694X $ - see front matter *Design Studies* **30** (2009) 676–703
doi:10.1016/j.destud.2009.06.005

Following this view of concept generation as a casual and disorderly activity, it is unclear how an automated conceptual generation process would benefit this phase of design. Design automation is clearly useful in later stages of the design process where a to-be-designed artifact accrues numerous parameters or degrees-of-freedom. Automated methods such as optimization provide a useful framework for managing and determining details of the final designed artifact. However, such structured methods would seem to be incongruous with the unstructured nature endorsed in the previous view of concept generation. We can argue that concept generation is limited to the experiences or creativity of the designers present at this stage of the process. The more experiences or past design knowledge available to the designer, the more variability will exist in the concepts that are generated. On the other hand, more experiences could reduce creativity in that the psychological bias of past design knowledge would prevent the team from finding truly creative solutions.

As a results of this standoff, we have developed an automated concept generation method to help aid designers in coming up with new solutions. The basis of the method is that knowledge is extracted from past designs, stored as procedural rules, and then employed in building new design concepts. This is accomplished by leveraging an expanding online repository of electromechanical products from which design knowledge is systematically extracted and integrated into a computational framework. The framework includes representations that capture designs at two levels of abstraction (the functional level and the component level) and, it allows the formulation of the design knowledge as procedural rules that depict the mapping between these two levels. The rules created from the repository are then used in a computational search process that works with a designer in navigating the design space to create conceptual design configurations from detailed specifications of product function. Using the developed design tool, a design team can generate many feasible concepts, which then can automatically be evaluated and ranked based on a variety of design objectives.

In this paper, we study how this automated approach (Kurtoglu et al., 2005a,b) benefits designers during concept generation. The objective of this research is to provide initial evidence that an automated tool can support the early concept generation process. The primary domain of the study is design engineering, however, the experimental methodology is more general in principle and can be used in a similar fashion in other areas of design to evaluate the effects of design automation.

1 Organization of the paper

In the following sections, we will present the automated concept generation tool and the details of the experimental approach. In Sections 2 and 3 of this paper, the related work and prior work by the authors will be summarized.

In Section 4, the details of the experimental method are discussed, followed by a presentation of the results in Section 5. The paper concludes with a summary of the results and the future directions of this research (Section 6).

2 Background

In this section, an overview of developing the grammar-based computational design framework is presented. This development involves three major processes: design knowledge extraction via empirical product analysis, representation of design knowledge, and computational design synthesis. (These processes constitute our existing work. An in-depth description of the three processes can be found in our previous articles (Kurtoglu et al., 2005a,b; Kurtoglu, 2007)). In the first process, product information is accumulated by means of product dissections, and empirical product analysis. The central part of this process involves capturing the function-based and configuration-based product information. The second process is concerned with the representation of the captured information in an easily computable format. Accordingly, two graph-based representations are used to model product information. Finally, the third process deals with creating conceptual design solutions. Initially, a design problem is put forth as a functional design specification consistent with the language used during the capture of the function-based product information. By using and combining existing product information, this design specification is converted to a configuration-based product description which is presented to the designer as a final conceptual solution. The general flowchart of the developed computational synthesis approach is shown in Figure 1. Each process is briefly described in the following sections.

2.1 Knowledge extraction via empirical product analysis

In establishing the concept generation approach, a rigorous method has been developed in extracting design knowledge and organizing the information so that it can be efficiently indexed, searched and retrieved. The first step in knowledge extraction is the dissection of products. The products are analyzed in terms of their functionality, components, and the relationships between product functions and components. Each product is completely dissected and documented by creating a detailed list of each component's attributes such as part number, quantity, part functions, energy flows through the components, physical parameters, etc. The design knowledge derived from product tear-downs is stored and organized in a web-based design knowledge repository, managed at the University of Missouri-Rolla. The knowledge contained in the repository is steadily expanding and currently includes detailed information on over 100 consumer products (including power tools, kitchen appliances and other household products) (Bohm and Stone, 2004). The grammar-development leverages the repository data to extract a design grammar language that captures the relationship between specific functions and the components that are used to fulfill them.

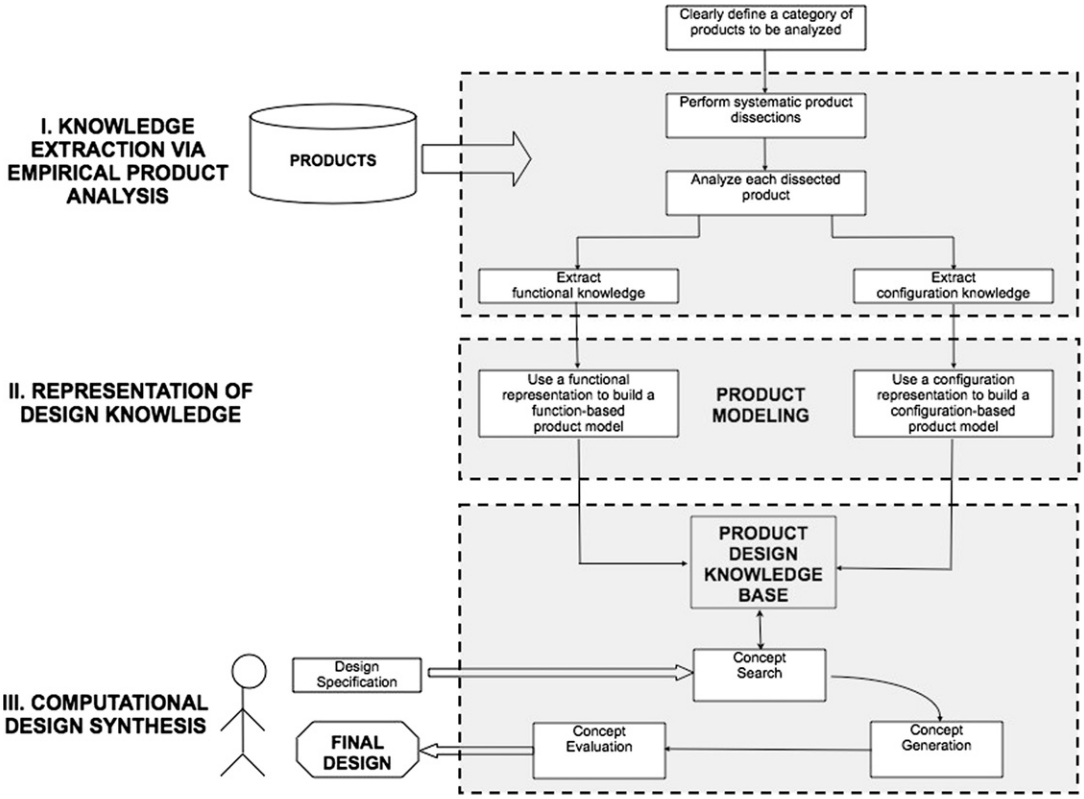

Figure 1 General flowchart of the computational synthesis process

Deriving uniformity and consistency in representing the functional and component design knowledge is essential to the development of the method. Towards that goal, two standardized vocabularies are utilized. At a functional level, the *functional basis* language is used (Stone and Wood, 2000). This standardized lexicon of terms define a comprehensive set of function and flow names at three levels of abstraction to support both high and low level functional modeling of a system. For example, the function of an electric motor is to "convert electrical energy to mechanical energy". Similar to the functional basis in principle, the *component basis*, provides a standardized set of component names (Kurtoglu et al, 2005a,b). Accordingly, each artifact is classified under a specific component name that can be thought of as a generic abstraction representing a component concept. The basis eliminates artifact redundancies that may not be immediately evident due to variations in user-dependent artifact naming. The current component basis contains 135 distinct fundamental component concepts. (Example component concepts include: acoustic insulator, agitator, electric motor, bearing, battery, belt, gear, housing, screw, etc).

2.2 Representation of design knowledge

Following the view of design that defines the process of concept generation as the transformation from function to form, the computational method described in this section converts a function-based model of a design solution to a configuration-based model. These models within the computational method are represented using two graph-based representations: *function structures* and *configuration flow graphs*. The former of these captures the design function, whereas the latter represents the configuration or topology of a design. These two graph representations constitute the foundation for the graph grammar language that the conceptual design tool is built upon.

A function structure (Pahl and Beitz, 1996) is a graphical representation of the decomposition of the overall function of a product into smaller, more elemental sub-functions. The sub-functions are connected by flows that are of type energy, material and signal. Overall, a function structure represents the transformation of input flows into output flows at the system level. Otto and Wood (2001) and Kurfman et al. (2001) put forth a method to build a function structure starting from the customer needs. Obtaining the customer needs, the generation of a black box model, the creation of function chains for each input flow, and the aggregation of function chains into a function structure are the sequence of steps that lead to the construction of a function structure. In order to attain a repeatable formation of function structures, the aforementioned functional basis (Stone and Wood, 2000) is used as a standard vocabulary during the construction of function structures.

A Configuration Flow Graph (CFG), on the other hand, is a graph representation of how the *functional* components are connected. In a configuration flow graph, nodes of the graph represent product components, whereas arcs represent energy, material or signal flows between components. For flow naming, the functional basis terminology (Stone and Wood, 2000) is adopted, while the components of the graph are named using the standard names of the component basis (Kurtoglu et al., 2005a,b). The configuration flow graph is a specific implementation of what some loosely define as the topology or the configuration of a product. The graph is also similar to an exploded view in that components (often drawn isometrically) are shown connected to one another through arcs or assembly paths. Figure 2 shows an example of a configuration flow graph (and a function structure) along with an exploded view for an electric toothbrush product. As can be seen from the figure, components that are present in a design, their connectivity, and physical interfaces between a design's components can be captured using a configuration flow graph. For example, by looking at the graph, one may conclude that the design is composed of a 'motor', which is actuated by a 'switch', and which drives a series of mechanical power transmission devices including a 'shaft', a 'coupler', and a 'link'.

2.3 Computational design synthesis

In this section we present the graph grammar approach for creating new design configurations from functional requirements. Graph transformation systems, or graph grammars, reside in graph theory research as a way to rigorously define mathematical operations such as addition and intersection of graphs. These approaches capture the transitions or the production rules for creating a solution, as opposed to storing the solutions themselves. Accordingly, an artifact's development from its inception to its final configuration is considered as a series of modifications. The initial specification can be represented as a simple graph in which the desired inputs and outputs are cast as arcs and nodes of the to-be-designed artifact. From this initial specification, the design process can be viewed as a progression of graph transformations that lead to the final configuration. This interpretation of the design process makes graph grammars very suitable for computationally modeling the open-ended nature of conceptual design, where designers explore various ideas, decisions and modifications to previous designs to arrive at feasible solutions.

Graph grammars are comprised of rules for manipulating nodes and arcs within a graph. The rules create a formal language for generating and updating complex designs from an initial graph-based specification. The development of the rules encapsulate a set a valid operations that can occur in the development of a design. Through the application of each grammar rule the design is transformed into a new state, incrementally evolving towards a desired solution. The rules are established prior to the design process and capture a certain type of design knowledge that is inherent to the problem. In the application presented here, the rules represent how functional requirements can be addressed by the use of certain individual or a group of components. To define the rules, an existing product's configuration flow graph (CFG) and function structure are built. Then, the mapping between the two graphs is captured and each potential mapping is defined as a ''grammar rule''. This procedure is illustrated schematically in Figure 3. In reality, each rule captures a design decision that shows how a functional requirement was transformed into an embodied solution in an actual design. (Currently, there are over 200 rules in the database).

The foundation of the grammar-based synthesis approach is to use this rule set in order to perform a graph transformation of the initial function structure of a to-be-designed artifact into a configuration flow graph. Accordingly, the grammar rules are defined to add components to the configuration flow graph that maintain a valid connection of components as well as meet specific function requirements specified with the function structure. Each of the rules developed are modeled after basic grammar conventions where rules are compromised of a left-hand side and right-hand side. The left-hand side contains a sub-graph that must be recognized in the function structure of the to-be-designed artifact and the right-hand side depicts how the design is transformed to a new configuration by the addition suitable components. The basic generation process for a set of rules is to first *recognize* which rules have left-hand sides that match the input specification of the design problem, then

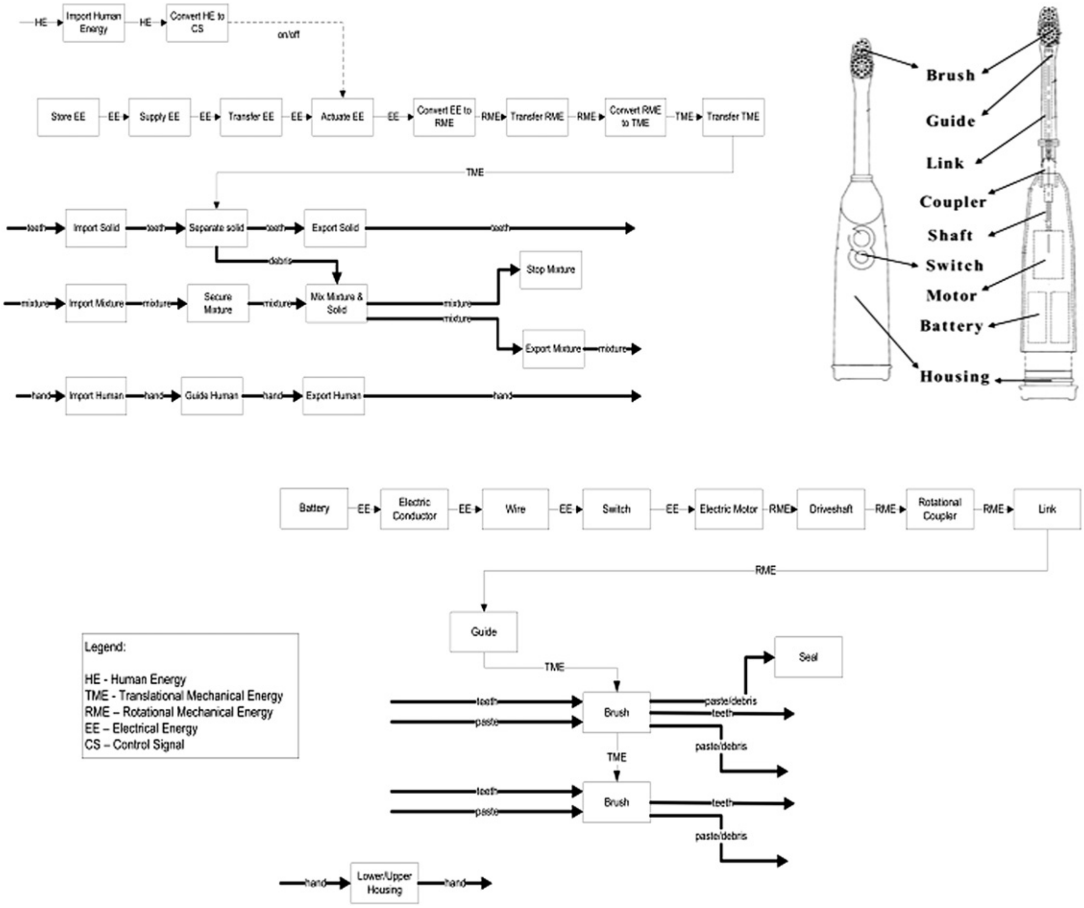

Figure 2 The function structure (top left), the exploded view (top right), and the configuration flow graph (bottom) of an electric toothbrush product

choose one of these rules, and finally to *apply* the rule as a step towards constructing an updated configuration.

The usage of a design grammar helps to generate a wide range of solutions by altering the way the rules are applied. The grammar affords a representation of the design space as a tree of solutions built from an initial specification as shown in Figure 4. Each transition in the tree corresponds to an application of a rule, thus incrementally building a final solution which is represented as one of the leaves of the tree. The result of rule applications generates a design space that requires navigation techniques to enable search for a desired or optimal solution. The issue of implementation of the grammar then becomes one of controlled searches through this space of solutions. The search process gives the designer the potential to explore a large number of alternative designs which include many alternatives that might have been overlooked without the aid of a grammar, thus paving the way for possible innovative designs.

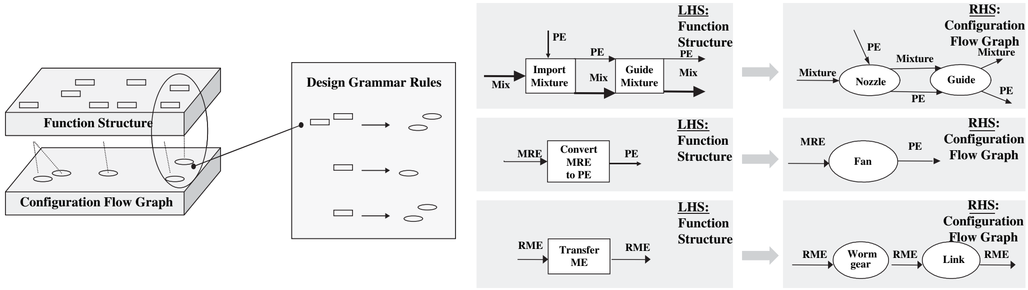

Figure 3 Illustration of rule derivation (left), sample rules (right)

At the end of concept generation, the created design solutions are evaluated and analyzed. This is accomplished by employing a strategy that selects and presents the designer a small-set of proposed solutions. These solutions are then rated by the designer based on specific design objectives and constraints. The designer's preferences are then used to automatically *guide* the search for finding 'best' configurations from the population of generated solutions (Kurtoglu and Campbell, 2007).

2.4 Software implementation and the graphical user interface

An algorithm has been created and programmed in C# using Microsoft Visual Studio. The design starts with the specification of the function structure of the product to be designed. This is accomplished through a graphical user interface that allows the designer to quickly draft a function structure. Figure 5 shows the user interface with the initial function structure (partially drawn) for a product. The program then executes the rules based on the input function structure. At any instant, there are a number of rules that can be applied and the recognition algorithm determines all of the possible rules by traversing the

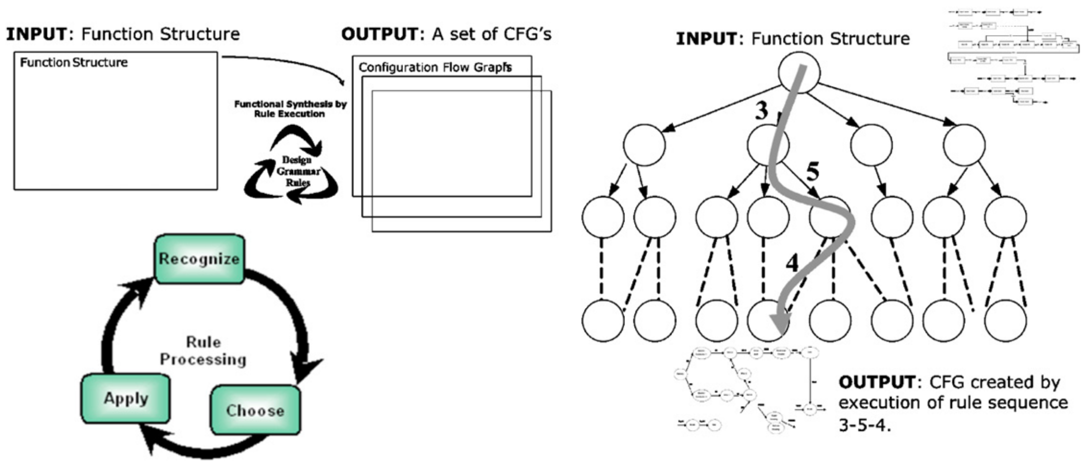

Figure 4 Illustration of rule processing via 'recognize-choose-apply' cycle. (left) A visualization of the search tree in building a CFG from a function structure. (right)

complete list of rules and checking each for applicability. Any rule that satisfies recognition conditions is then listed as a 'recognized' rule along with its corresponding location. Recognition is followed by the selection of the rule to be applied. In the current implementation, the rules that are actually applied are selected automatically as the design space is exhaustively searched for all possible configurations. The third and final step of rule processing is applying a selected rule. After the program makes a selection from the recognized rules list, it updates the function structure graph to a new configuration by replacing related sub-functions with their associated component(s) as described by the selected rule. The program manages the rules and their applications until no further rules can be applied, thus terminating when a complete design configuration has been built for the given functional description. One such configuration is shown in Figure 5.

3 Related work

3.1 Idea generation techniques

Concept generation research has traditionally focused on developing methods that improve the quality and variety of concepts generated. These methods are often kept simple and efficient such that designers are not burdened by the details or limitations of the method. The most common concept generation method is known as brainstorming (Osborn, 1957). The term brainstorming is frequently applied to any idea generation technique. Brainstorming as a specific method requires a group of individuals to follow the basic rules of (1) avoiding criticism, (2) welcoming 'wild ideas', (3) building on one another's ideas, and (4) preferring more ideas than dwelling on specific ones. A more structured concept generation method can be found in the techniques known as C-Sketch (Shah, 1998) and 6-3-5 (Rohrbach, 1969). Both of these methods require six participants to independently create three ideas at a time in a series of five rounds. The added constraints of the method ensures that individuals participate equally which may be difficult to enforce in traditional brainstorming.

In addition to these group methods for concept generation, there are also some well accepted approaches that do not require a set of interacting designers. As an example, designing by analogy is a well accepted approach to arrive at novel design solutions. One novel approach for designing by analogy is to first generalizing the design problem to a set of functions (or a graph of functions as in the function structure representation). Then one can look for or conceive analogous products or components that perform the same set of functions (McAdams and Wood, 2000). Function-means trees and Morphological Analysis (Zwicky, 1969) are similar methods in which solutions to individual functional requirements are first sought and then synthesized together. Apart from these approaches, one widely used method is the Theory of Inventive Problem Solving (Altshuller, 1984). This method provides a tabulated representation of a large number of solution principles that have been extracted from existing

Input is cast as a "function structure"

Output is a "configuration flow graph"

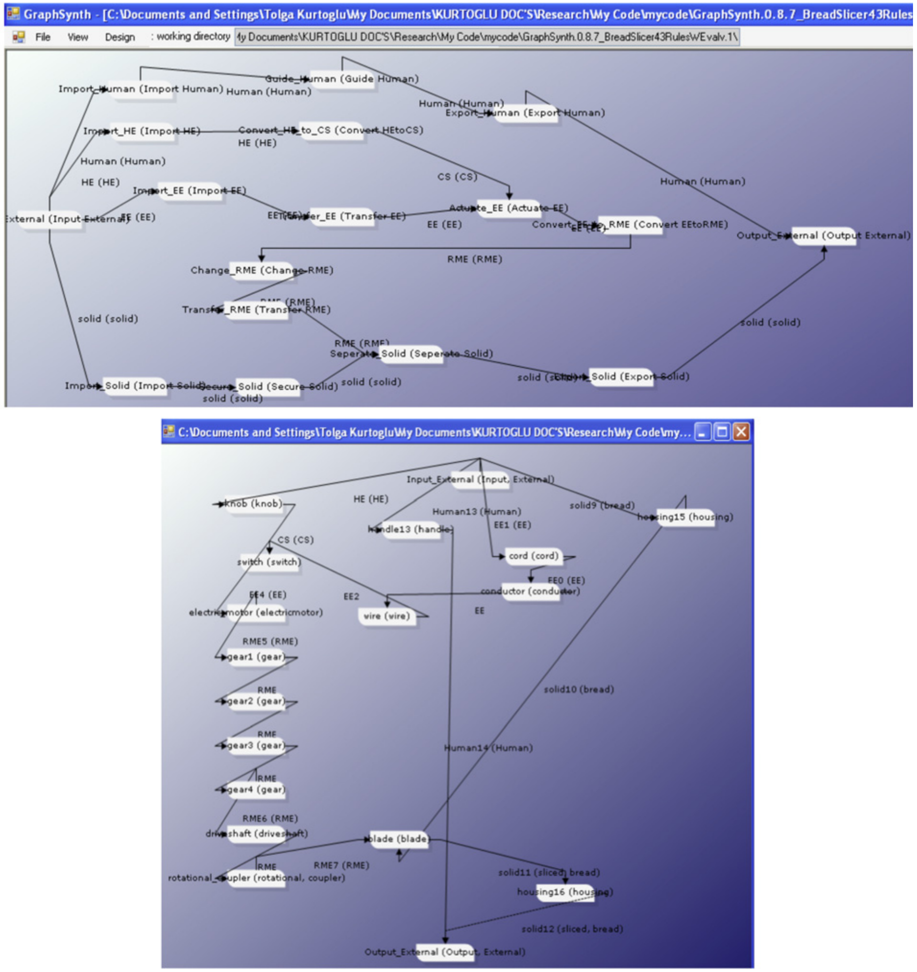

Figure 5 Graphical user interface of the developed automated concept generation software. The designer inputs the function structure of the system that is to be designed (top). A screenshot of the user interface showing one of the many designs after the completion of automated synthesis (bottom)

patents. Another approach is 'catalog design' where concepts are generated purely through browsing a catalog of physical elements (components, assemblies, etc.). The results are evidently limited by the breadth of the catalog; however, the benefit lies in the presentation of design knowledge that falls outside the designer's expertise memory (McAdams and Wood, 2000). The method that is developed in this research is similar to this design by catalog approach in that the computational search process is based on an online repository of past electromechanical designs.

3.2 Computational design synthesis

Various approaches have been attempted to solve conceptual design problems with computational methods. Two such methods are the agent-based system presented in the A-Design research (Campbell et al., 2000) and the catalog design method used in Chakrabarti and Bligh (1996). In both these approaches, input-output characteristics of components are used to synthesize a system level design by using components as building blocks and by integrating them according to design requirements.

Other methods to computational design synthesis include representations that manage and manipulate functional descriptions which are later converted into configurations of components. Bracewell and Sharpe (1996) used a method based on energy flows as a foundation in their Schemebuilder tool to automatically explore alternative conceptual schemes and appropriate allocation of function between electromechanical components. Following a similar function-based approach, Bryant, et al. (2005) developed a concept generation technique that utilizes a repository of existing design knowledge and a set of matrix-manipulation algorithms. In this research, function–component matrices are used to capture and store information about what particular components can be used to solve specific functions and component–component matrices are used to encode physical compatibilities between components. Matrix algebra is then employed on these two matrices to construct a final product matrix describing the complete solution space.

In recent years, engineering researchers have discovered that graph grammars provide a flexible yet ideally structured approach to engineering design synthesis (Cagan, 2001). As described in the previous section, the concept of a grammar is that a set of rules is constructed to capture specific domain knowledge about a certain type of artifact. The development of these rules encapsulate a set a valid operations that can occur in the development of a design. These techniques create a formal language for generating and updating complex designs from a simple initial specification, or seed. In developing the aforementioned computational tool, we combine the formal function-based synthesis method (Pahl and Beitz, 1996) with graph representations developed to facilitate the formulation of a graph grammar language for creating design configurations. Accordingly, we use a functional description of a product as a seed and seek multiple configuration solutions that address the functional requirements.

4 Experimental research method

The objective of the research reported in this paper is to study the value of the aforementioned computational approach to design synthesis. Accordingly, in this section, we describe the experimental approach developed to assess the effects of using the 'design configuration graphs' generated by the automated conceptual design tool as a conceptual design aid. This study complements our prior work (described in Section 2) in the field of design automation and aims to address some of the fundamental research issues concerned with understanding whether the configuration graphs benefit designers to actually help create better ideas and thus better concepts. More specifically, we seek to gain insight into the following research questions:

- How *useful* are configuration flow graphs in creating design concepts?
- To what extend can the *variety* of generated solutions be improved by the use of configuration flow graphs?
- Does the use of configuration flow graphs produce more *complete* concepts?
- Is there an interaction between the *novelty* of ideas generated and the use of configuration flow graphs as a design aid?

We attempt to answer these research questions in the following sections. These questions can be generalized beyond the analysis of the use of configuration flow graphs for concept generation and can be posed as more generic questions towards understanding the overall effects of computational tools on human conceptual designing. Such an approach, however, would require many larger-scale studies by the engineering design community. In this research, we develop an experimental method and follow it to answer the aforementioned questions within the scope of using the configuration flow graphs in order to assess the potential of the development of our computational design tool.

In the following sections, first, we present the two design problems selected to constitute the empirical basis for the development of our experimental method. Next, we describe our experimental procedure that aims to compare conceptual sketches created by engineering designers using the configuration graphs to conceptual sketches created by designers not using the graphs. Finally, we outline our evaluation metrics and evaluation procedure.

4.1 Design problem descriptions

Two design problems are selected to mimic industry level design challenges. The first problem is to design a device that caps bottles. The second problem is to design a soda making device for use as a kitchen appliance. The problems are selected such that their primary domains are different (mechanical, fluidic) and that there is minimal overlap in their functional requirements.[1]

The problem descriptions that are given to the designers are as follows:

Design Problem 1: Bottle Capping Device: 'Design a machine that registers a bottle to a capping station, caps it, and allows somebody to retrieve the capped bottle from the device. Please do not limit your design to a particular bottle, cap geometry. You can assume that you have control over the specifics of both these system inputs and how they should interface with each other'.

Design Problem 2: Soda Maker: 'Design a device that takes water, sodium bicarbonate (gas), and soda flavor syrup as input and mixes them into a soda drink. The device is targeted as a home type kitchen appliance. The inputting of the water can be accomplished through a standard kitchen faucet. Please assume that the soda flavor syrup is available in a separate container that can be poured into the device you are designing, and the sodium bicarbonate is contained in a canister that can safely transfer sodium bicarbonate into the system'.

Along with a design description, a conceptual function structure of the product to be designed is also given to the designers for each problem. These function structures are constructed by the experimenters after studying the problem descriptions and particular design requirements. One might think that providing the functional graphs constitutes a constraint on problem definitions, however this is a necessary step for establishing a consistent comparison basis for the experiment. By providing a functional model along with the problem descriptions, we are able to eliminate potential variation due to varying interpretations of functional requirements. Figure 6 shows the two function structures presented to the designers.

4.2 Participants

The participants are graduate students from the University of Texas at Austin. These students, a total of sixteen, average 12.8 months of professional engineering experience (out of school) ranging from no experience (3 participants) to 4 years of experience. They all have recently studied in a graduate design methodology course that teaches a number of idea generation techniques including Brainstorming, Mind Mapping, 6-3-5, patent searching, TIPS, and design by analogy. In the same course, they also learn about how to create function structures using the functional basis, thus the participants are proficient in reading and interpreting function structures such as the ones shown in Figure 6.

4.3 Experimental procedure

The main goal of the experiments is to assess the value of the configuration flow graphs as a conceptual design aid. To explore the effect of the use of

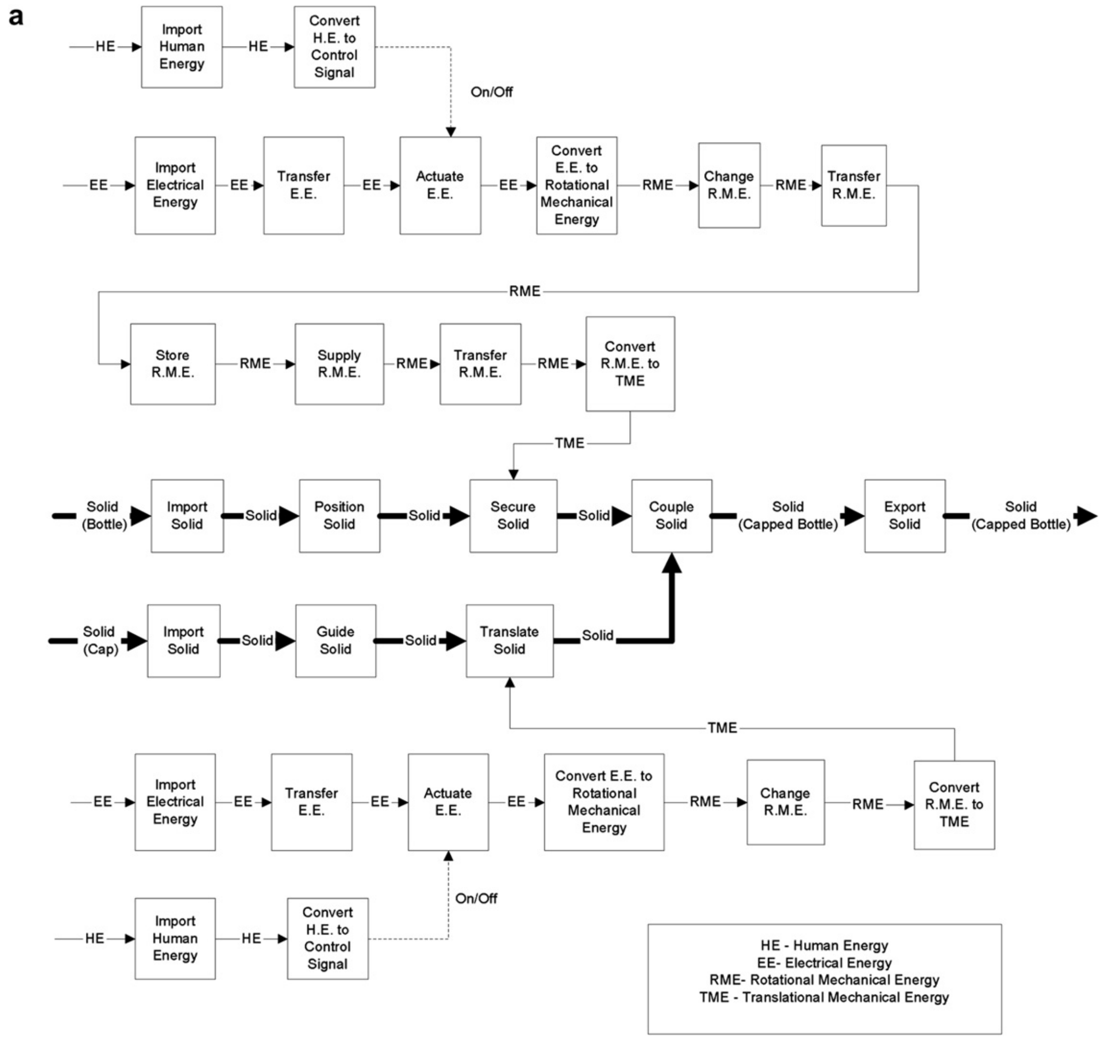

Figure 6 (a) The function structure for the ‘bottle capping device’. (b) The function structure for the ‘soda making device’

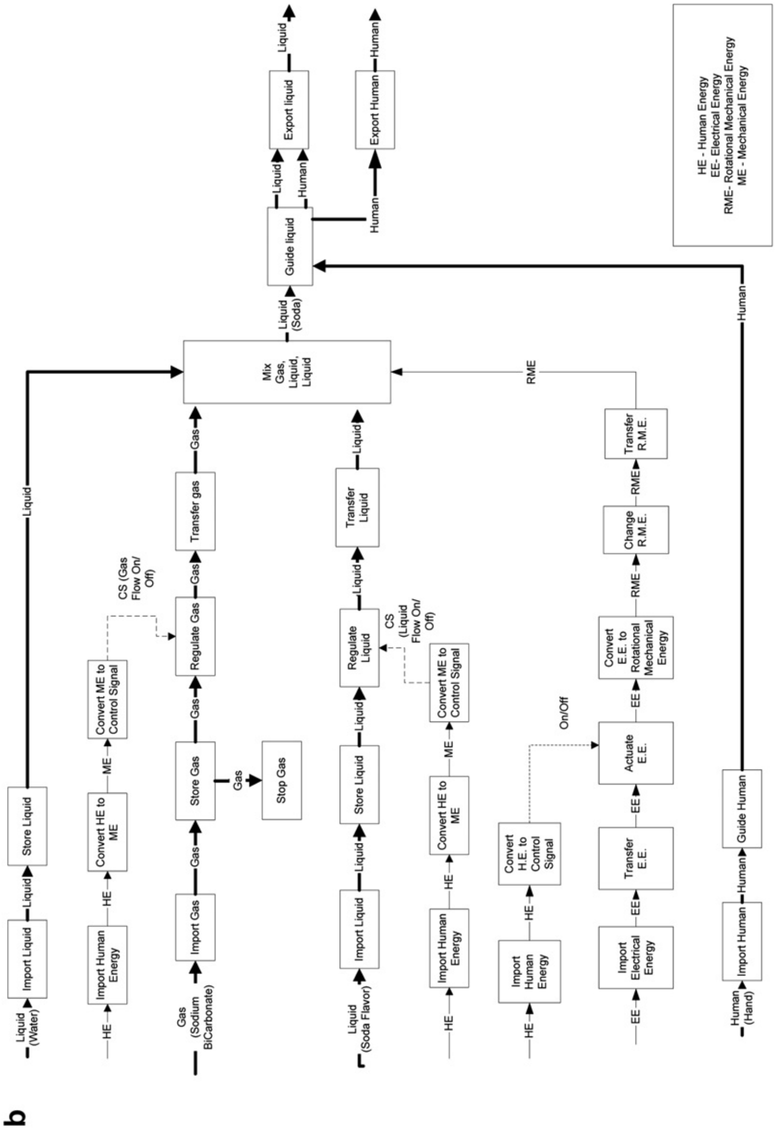

Figure 6 (continued)

configuration flow graphs, an experiment is conducted. Accordingly, a set of sixteen participants is randomly divided into four experimental condition groups. Each group includes 4 designers who individually generate concepts to the two design problems. For each individual, one of the design problems is accompanied by a set of configuration graphs that is presented as catalog-like reference material that the participant may choose to use as a design aid for concept generation. For the other problem, no design configuration graph is provided. The order of design problems is also randomized and assigned to each individual. Table 1 illustrates the summary of the four experimental conditions.

The design configuration graphs that are given to the participants are automatically created by the authors using the computational synthesis tool. A set of configuration graphs are then manually selected for each problem and given to the participants. This set includes six configuration graphs for the 'bottle capping' problem and three configuration graphs for the 'soda maker' problem. Figure 7 shows sample configuration graphs out of this combined set of nine.

The experiment is conducted over a two week period. The participants are asked if they have participated in any form of concept generation exercise for the given problems prior to the session and are excluded from the experiment if they have done so. At the beginning of each session, the experimenter read a script of instructions explaining the experimental procedure. The instructions include a description of the problem, a conceptual function structure of the problem, and basic guidelines for the prescribed method. The guidelines outline that the participants are expected to generate as many solutions in the allotted time as possible. The participants are also told that they should represent their ideas using sketches and/or written words and that their generated concepts should address the majority of the functional requirements specified by the given function structure. For sessions including configuration graphs as a design aid, there are additional instructions explaining what a configuration graph represents followed by the selected set of configuration graphs for the particular concept generation session. In these sessions, participants are asked to study this additional material before generating concepts and are instructed that they *may* use concepts from the presented configuration graphs during their exercise. The sessions last approximately 50–60 min

Table 1 Experimental conditions

	Session I	*Session II*
Group I (4 designers)	Bottle capping without CFG's	Soda maker with CFG's
Group II (4 designers)	Bottle capping with CFG's	Soda maker without CFG's
Group III (4 designers)	Soda maker with CFG's	Bottle capping without CFG's
Group IV (4 designers)	Soda maker without CFG's	Bottle capping with CFG's

with an equal 45 min allocated to idea generation whether or not the session includes the use of configuration graphs. Each session is followed by a post-session questionnaire that asks questions about the experiment and the use of configuration graphs.

4.4 Evaluation metrics

The development of proper metrics for concept evaluation during engineering problem solving is an emerging field of study (Otto and Holtta, 2004). Shah et al. (2000, 2003) propose a set of four metrics specifically developed for the evaluation of idea generation methods used for conceptual design. These four metrics are defined based on how well a concept generation method expands a design space, and how well it explores the resulting space.

a

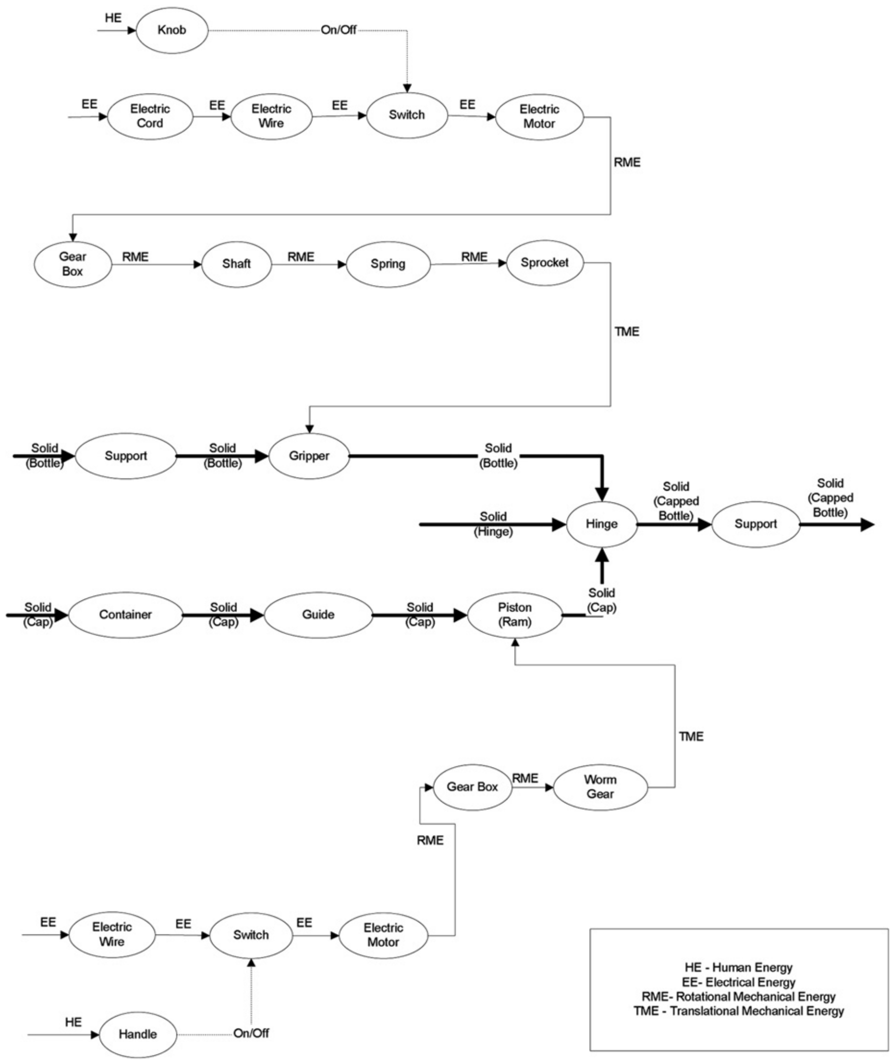

Figure 7 (a) A configuration flow graph for the 'bottle capping device'. (b) A configuration flow graph for the 'soda making device'

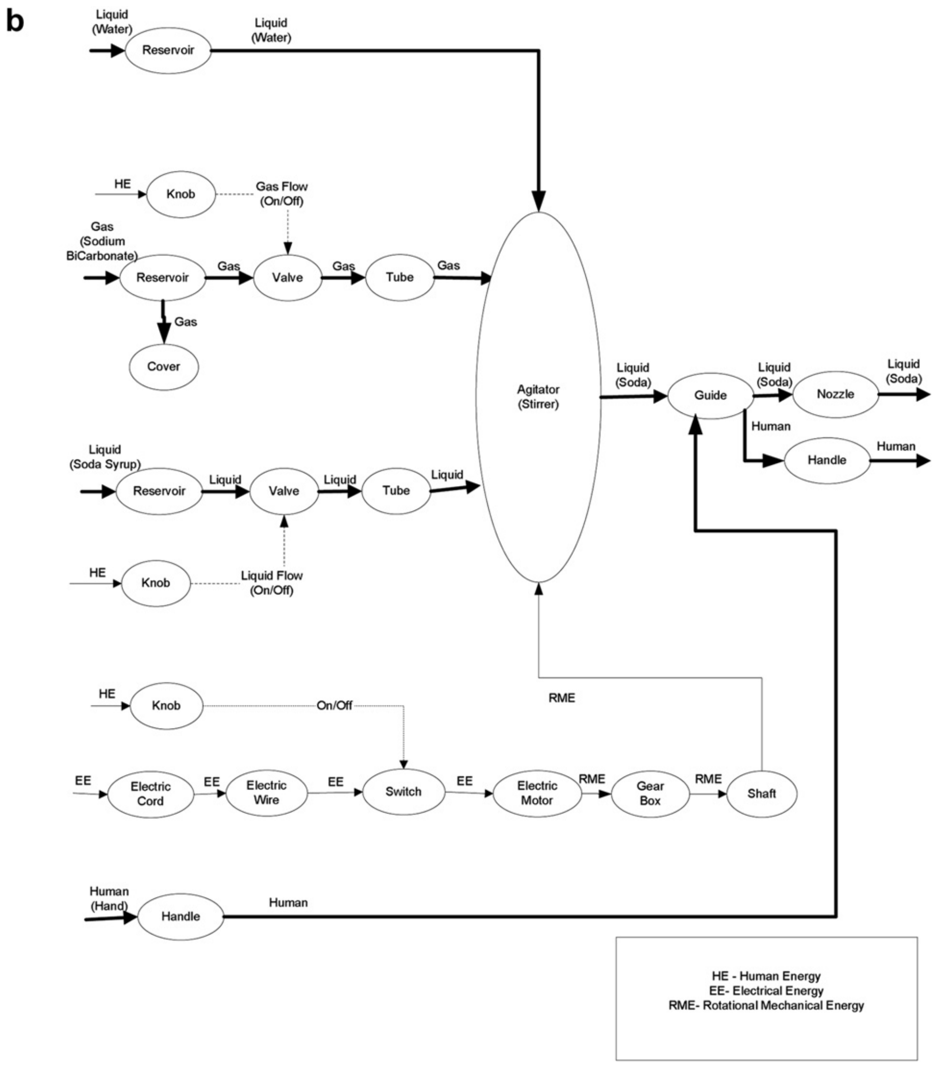

Figure 7 (continued)

In this study, we adopt two of these four metrics (*novelty, and variety*) and add one of our own (*completeness*). The definitions of these metrics are as follows:

- *Completeness:* is a measure of the level indicating how much a concept variant addresses the sub-functions depicted in the function structure.
- *Novelty:* is a measure of how unusual or unexpected an idea is as compared to other ideas (Shah et al., 2000).
- *Variety:* is a measure of the explored solution space during idea generation (Shah et al., 2000). This is measured across a set of concepts produced by a participant rather than for a single concept like novelty and completeness.

4.5 Evaluation procedure

Two doctoral candidates and a faculty member were selected as judges from the Manufacturing & Design Division of the Department of Mechanical Engineering at the University of Texas at Austin. All three evaluators are experts in the areas of functional modeling and concept generation and all have given lectures on related subject matters. The judges independently evaluated the resulting concepts[2] according to the metrics completeness, novelty, and variety. The first two of these metrics are evaluated individually for each concept generated, whereas the last metric is evaluated on a set basis for each participant.

To perform the evaluations, all three judges were given a shuffled stack of concept sketches without knowing which designs were created with the aid of the configuration graphs and which without. Each judge is instructed to go through all of the solutions and evaluate them according to the guidelines summarized. For concept scoring, an ordinal 1 (low)–7 (high) scale is used. Before running statistical analysis on the evaluation data, final scores are computed by averaging the individual scores of the three judges.

5 Experiment results and discussion

5.1 Summary of the generated data and the statistical analyses performed

A total of 125 (56 for ‘Bottle Capping Device’, and 69 for ‘Soda Making Device’) conceptual sketches are created by the 16 participants with varying degrees-of completeness, novelty and variety. (Six of these sketches are presented in Figure 8.). A number of statistical analyses are run on the generated data. These are summarized below:

- Inter-agreement between judges: To monitor agreement between judges’ scores, pair wise Spearman rank-order correlation coefficients (Clark-Carter, 1997) are calculated first. Low correlations between judges indicate that the method for measuring the metric is not reliable and therefore the metric is not reliable. For unreliable metrics, no further analysis is completed.
- Interdependency of the evaluation metrics: The interdependency of the three evaluation metrics (completeness, novelty, and variety) are analyzed by calculating the Spearman correlation coefficients between the metrics.
- Problem order effects: The problem order effects are analyzed. Even though the problem order was counter-balanced in the experimental setup, a statistical analysis is run to verify that problem ordering did not have any effect on the outcome. Accordingly, the data is analyzed using a three-factor ANOVA[3] with repeated measures on the 2nd and 3rd factors and a Mann–Whitney *U*-test.
- Normalizing the data: The generated data is pre-processed for further statistical analysis. This is accomplished by normalizing the data for each participant by averaging the individual scores of the concepts generated by that

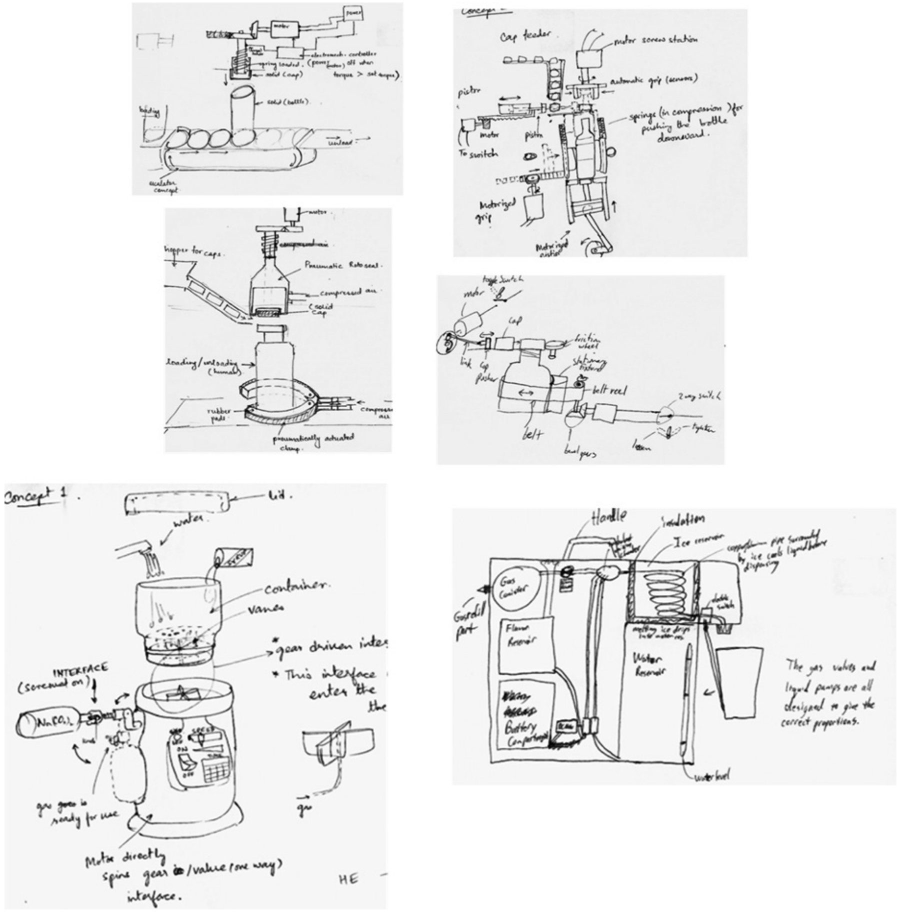

Figure 8 Sample sketches created by participants

participant. This ensures that the analysis is not biased towards participants who create greater number of concepts. Participants produced anywhere from two to seven concepts for the soda maker problem and one to six concepts for the bottle capping problem. This data is presented in Table 2, Figure 9, and Figure 10.

- Variety', 'Novelty', and 'Completeness' analysis of the combined data: The normalized data is analyzed for the three metrics of 'variety', 'novelty', and 'completeness'. All the statistical analyses are performed using Mann–Whitney *U*-tests (Clark-Carter, 1997) through SPSS statistical analysis software. Mann–Whitney *U* is used rather than t-test, ANOVA or MANOVA because the data is ordinal and the judges scores are not normally distributed (Figure 11). (Mann–Whitney *U* is the non-parametric equivalent of a *t*-test and compares the ranked scores of the two groups.)
- Variety', 'Novelty', and 'Completeness' analysis of the 'top' and the 'top two' ideas from each participant: In addition to the initial analysis on the

Table 2 Distribution of experimental data

		Soda maker			Bottle capping
		Completeness	*Novelty*	*Variety*	*Completeness*
With CFG	25 percentile	3.81	3.60	3.83	2.44
	Median	4.17	3.74	4.50	3.92
	75 percentile	5.17	4.17	5.08	4.25
Without CFG	25 percentile	3.31	2.87	2.58	3.13
	Median	3.53	3.59	3.50	3.54
	75 percentile	4.20	4.24	4.33	4.18

normalized data, the top idea and the top two ideas, as determined by the sum of the completeness and novelty scores, from each participant are also analyzed. For the bottle capping device, the top two ideas are not analyzed since one participant produced only a single concept. Again, all the statistical analyses of this step are performed using Mann–Whitney *U*-tests (Clark-Carter, 1997) running on SPSS statistical analysis software.

5.2 Statistical analysis results

The results of the statistical analyses are summarized below:

- Inter-agreement between judges: For the soda maker problem: completeness [0.52–0.57], novelty [0.44–0.63], and variety [0.44–0.78]. For the bottle capping device: completeness [0.50–0.75], novelty [0.17–0.24], variety [−0.07 to 0.33].
- Interdependency of the evaluation metrics: The Spearman correlation coefficients between the three metrics are tabulated in Table 3.

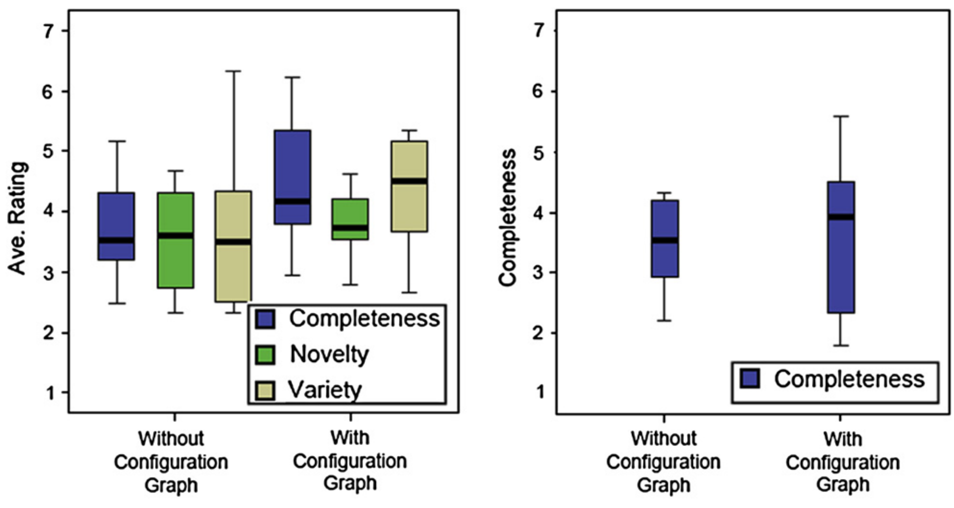

Figure 9 Box plot summary for bottle capping device (right), and for soda maker device (left)

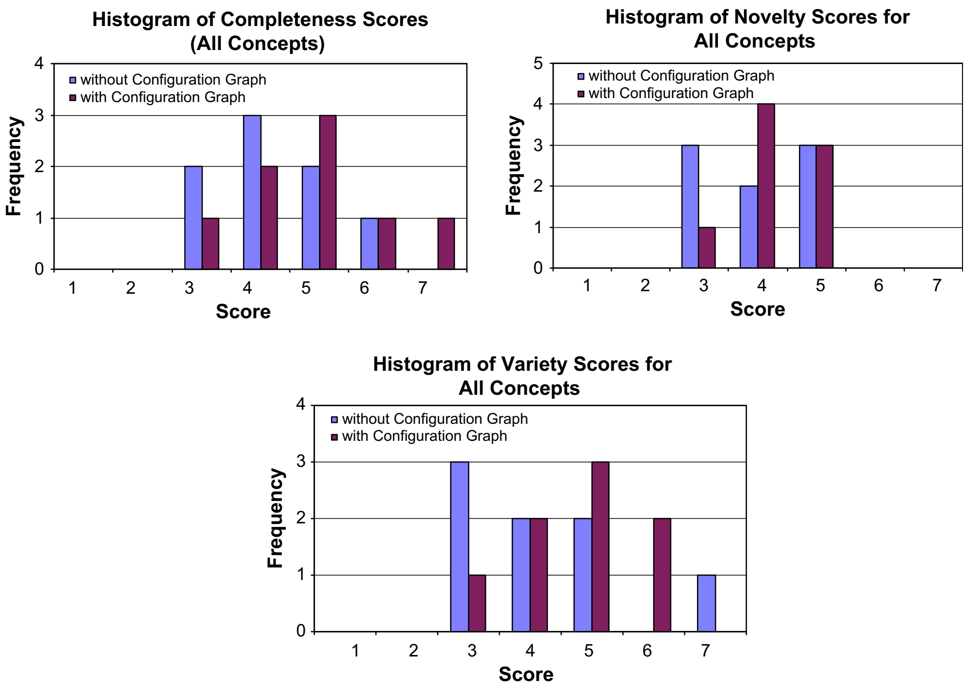

Figure 10 Histograms showing the score distributions for the soda maker device

- Variety', 'Novelty' and 'Completeness' analysis: Analysis results are summarized in Tables 4–7.

5.3 Discussion of analysis results

- Inter-agreement between judges: For the soda maker problem, an acceptable level of correlation occurs for all three metrics. For the bottle capping device, on the other hand, only the completeness metric yields adequate

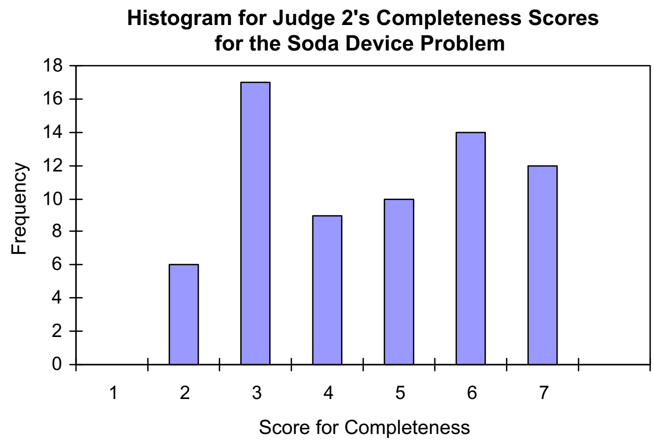

Figure 11 In general the judges scores do not follow a normal distribution

Table 3 Spearman correlation coefficients between evaluation metrics (for the soda maker)

	Spearman correlation coefficient		
	Novelty and completeness	*Novelty and variety*	*Completeness and variety*
Judge 1	0.12	0.25	0.26
Judge 2	0.28	0.49	0.10
Judge 3	0.14	0.36	0.13
Ave. of 3 Judges	0.18	0.77	0.16

correlation. The analysis on inter-agreement between judges yields that other two metrics for this problem did not have good correlation between the judges and are therefore excluded from further analysis. This also suggests that more investigation is necessary to understand why the judges were in disagreement in their variety and novelty scores for the bottle capping problem.

- Interdependency of the evaluation metrics: Based on the results tabulated in Table 3, there is no correlation between the metrics for the individual concepts thus indicating that the metrics are independent. However, there is correlation between a concept set's (i.e., all concepts generated by an individual participant) average novelty and its variety. This does not imply the metrics are correlated instead it implies concept sets containing very novel ideas tend to have greater variety overall. This is not surprising since novel ideas tend to be very different from common ideas and it is easy to think of the common ideas. The concept sets with a high degree of variety are likely to contain a combination of very novel ideas and very common ones.
- Problem order effects: According to the results, there is no significant effect on the results (all the three metrics) due to the order in which the problems are presented to participants.
- Variety', 'Novelty', and 'Completeness' analysis: The objective of this study was to investigate the implications of integrating an automated design tool into the concept generation process which is considered to be a highly creative and human-oriented activity. The aim was to provide designers with a tool that would benefit them during idea generation by presenting them information that may fall outside their personal experiences (or immediate memory) and to enable them to readily reuse existing design knowledge. Accordingly, the grammar-based concept generation approach gives the designer the ability to explore a large number of alternative designs which

Table 4 Mann−Whitney *U*-test results for normalized data

	Soda maker			*Bottle capping device*
	Completeness	*Novelty*	*Variety*	*Completeness*
Mann−Whitney *U*	19.0	28.0	19.0	31
P (significance)	0.19	0.72	0.19	0.96

Table 5 Mann–Whitney *U*-test results for the best idea from each participant

	Soda maker		*Bottle capping device*
	Completeness	*Novelty*	*Completeness*
Mann–Whitney *U*	14.5	27.0	28.0–31.5[b]
P (significance)	0.07[a]	0.65	0.72–0.96[b]

[a] Statically significant.
[b] For two participants there is a tie for the best idea therefore all combinations are evaluated and the range is shown.

include many alternatives that might have been overlooked without the aid of a grammar, thus paving the way for possible innovative designs.

On the other hand, providing designers with such design aids, do carry some risks. First, they may hinder creativity, and second, they may cause design fixation that prevents designers from exploring truly novel ideas. The metrics 'variety', 'novelty' and 'completeness' are defined considering these equally valid arguments. Based on the tabulated results of Tables 4–7, the analysis show configuration graphs having a potential to enhance the idea generation process meriting further development and study for their use during concept generation process. The following observations stand out from the analysis results:

- Configuration graphs have the potential to help designers create more complete concepts early in the design process by allowing them to approach the problem in a more systematic, function oriented way. The ideas from each participant for the soda maker are statistically more complete when participants have access to configuration graphs than when they do not (Figure 10, and Tables 4–6).
- The median score for the novelty metric for the soda maker concepts are higher with the aid of configuration graphs, albeit the Mann–Whitney *U*-Test shows no statistical significance (Table 2, Figure 10). This means that contrary to the claim that stated the configuration graphs cause fixation on certain ideas during concept generation, no statistical evidence was found that suggests the use of the graphs hinders a designer's creativity. Even when only the most novel concepts from each person are analyzed, there is no statistical difference between the groups in their 'novelty' score (Table 7).
- The median score for the variety metric for the soda maker concepts are higher with the aid of configuration graphs, albeit the Mann–Whitney

Table 6 Mann–Whitney *U*-test results for the top two ideas from each participant

	Soda maker	
	Completeness	*Novelty*
Mann–Whitney *U*	77.5	93.0
P (significance)	0.06[a]	0.20

[a] Statically significant.

Table 7 Mann–Whitney *U*-test results for the most novel idea from each participant

	Soda maker	
	Completeness	*Novelty*
Mann–Whitney *U*	17.0	19.0
P (significance)	0.13	0.20

U-Test shows no statistical significance. (Table 2, Figure 10). This is an encouraging result considering the fact that the current database includes information extracted from just over a hundred products. As the knowledge base grows, it is expected for the configuration graphs to have a stronger influence on the variety scores.

5.4 Survey results

To better understand the usefulness of providing designers with configuration flow graphs, a post-session questionnaire asked the participants several questions concerning how they felt about using the graphs. The answers to two such questions are summarized in Table 8. According to the results, the majority of the designers liked the option of being provided the configuration graphs.

93.75% of the designers found the design configuration graphs easy to use (18.75% somewhat easy, 68.75% easy, 6.25% very easy). This means that the graph-based scheme was a good decision to represent configuration-based design information and that the modeling language used to construct the graphs was appropriate. On the other hand, a 100% of the designers found the configuration graphs useful, though to a varying degree. (37.50% somewhat useful, 43.75% useful, 18.75% very useful) Some designers complained about being constrained or fixated by the presence of the design configuration graphs. However, as mentioned in the previous section, no evidence was found that the graphs cause design fixation.

Table 8 Snapshot of the survey results

Survey results from concept generation sessions (16 participants)						
Question	Very difficult	Difficult	Somewhat difficult	Somewhat easy	Easy	Very easy
How easy was it to understand the design configuration graphs?	0.00%	0.00%	6.25%	18.75%	68.75%	6.25%
Question	Not Useful	Somewhat Useful	Useful	Very Useful		
How useful was the design configuration graphs in creating concepts to the given design problem?	0.00%	37.50%	43.75%	18.75%		

From an open-ended question on the survey that stated, 'Please make any additional comments you have about the concept generation exercise', numerous designers reiterated the fact that they found the configuration graphs to be beneficial. Some comments from the participants included: 'the design configuration graphs were helpful…', 'the design configuration graphs give you some place to start…', 'the configuration graphs definitely helped, and the solutions generated by the configuration graphs were very exhaustive and comprehensive.' In short, the survey results indicated that designers felt positive about using the configuration graphs. They found the graphs easy to understand and to their benefit in creating conceptual solutions.

6 *Conclusion and future work*

In this paper, we have presented a study performed to evaluate the effects of a recently developed automated conceptual design tool on concept generation. The computational tool is developed based on a method that leverages existing design knowledge by integrating design concepts from past designs together to create novel solutions. The design knowledge is captured through systematic dissection of electromechanical products and the construction of an online design knowledge base. Using this knowledge base, design rules are formulated that can then be reused to build new design concepts.

The development of the computational tool is intended to help designers choose suitable components for given functional requirements in a redesign or an original design situation. We test outputs from the developed computational tool to determine their value in improving the concept generation results. The experimental approach involves 16 test subjects each creating solutions to two separate design problems: one in which they use outputs from the computational method in the form of design configuration graphs, and one in which they do not. Three metrics are then used to evaluate the solutions generated: completeness, novelty and variety.

The data from the experiments support our hypothesis that the integration of automatically generated design aids such as configuration flow graphs into the concept generation process results in an improved idea generation performance. The designs obtained by using configuration graphs proved to be more complete, without any decrease in variety or novelty.

Since the study was designed to provide preliminary evaluation of the configuration graphs and their potential usefulness in the design process, there are a number of assumptions that need to be taken into account when interpreting these results. The idea generation was an individual not a team activity. The short 45 min time period for idea generation is a bit short for the complexity of the design problems but was long enough for many participants to run out of ideas. Also, this study only took a snapshot of the design process just before and just after the idea generation. This was done to set-up a reasonable experiment and to limit the noise that other phases in the design process may add to the experiment. However, the effects of this decision on the quality of the final product are not known. These areas are open for future research.

Other areas identified for future work include the expansion of the knowledge base, more human studies and improved metrics. Currently, we are studying a variety of other products to increase the number of components and solution principles in the knowledge base. The database is steadily expanding with the addition of new products. We expect this to positively influence the degree of variety and novelty of future designs. In addition, we are exploring ways to develop features that can be incorporated into the existing user interface helping designers to sort or rank generated design concepts or to retrieve designs with certain attributes. Such tools will automatically identify the most promising designs and present the designer with the information that better stimulate a designer's creativity.

The evaluation of the effectiveness of these new features remains the subject of future experimental studies. A complete evaluation of any computational tool requires well-designed experimental studies involving industrial practitioner preferably in a long-term, realistic industrial setting. As enhancements to the computational tool occur, additional human studies will need to validate its usefulness and to guide additional improvements. The results obtained from this initial study, however, support our automated approach to design synthesis as a promising new method that benefits designers, especially novice ones.

Acknowledgements

The authors would like to thank the National Science Foundation for supporting this work under grant award number IIS-0307665.

1. As represented by the sub-functions of their respective function structures.
2. A concept here is defined as a collection of ideas developed and synthesized together in order to address functional requirements given by the problem descriptions. Figure 9 shows six unique concepts created by the participants.
3. We have initially used ANOVA to analyze ordering effects, even though our data deviates from this test's assumptions. There is not a common approach for analyzing ordering effects of ordinal, non-normally distributed data (Gliner and Morgan, 2000).

References

Altshuller, G (1984) *Creativity as an exact science,* Gorden and Breach, Luxembourg

Bohm, M and Stone, R (2004) Product design support: exploring a design repository system, in *Proceedings of IMECE'04, IMECE2004-61746*, Anaheim, CA

Bracewell, R H and Sharpe, J E E (1996) Functional descriptions used in computer support for qualitative scheme generation—schemebuilder, *AIEDAM* Vol 10 No 4 pp 333–346

Bryant, C, Stone, R, McAdams, D, Kurtoglu, T, Campbell, M (2005) A computational technique for concept generation in *ASME 2005 International Design Engineering Technical Conference*, Long Beach, CA

Cagan, J (2001) Engineering shape grammars, *Formal engineering design synthesis* in **E K Antonsson and J Cagan** (eds) Cambridge University Press

Campbell, M, Cagan, J and Kotovsky, K (2000) Agent-based synthesis of electromechanical design configurations, *Journal of Mechanical Design* Vol 122 No 1 pp 61–69

Chakrabarti, A and Bligh, T P (1996) "An approach to functional synthesis of mechanical design concepts: theory, applications and merging research issues,", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* Vol 10 pp 313–331

Clark-Carter, David (1997) *Doing quantitative psychological research: from design to report* Psychology Press, Hove

Gliner, J and Morgan, G A (2000) *Research methods in applied settings: an integrated approach to design and analysis* Lawrence Erlbaum Associates, New Jersey & London 205

Kelley, T and Littman, J (2001) *The art of innovation: lessons in creativity from IDEO, America's leading design firm* DoubleDay, New York

Kurfman, M, Rajan, J, Stone, R and Wood, K (2001) Functional modeling experimental studies, in *Proceedings of DETC2001, DETC2001/DTM-21709*, Pittsburgh, PA

Kurtoglu, T (2007) A computational approach to innovative conceptual design, Ph.D. Dissertation, The University of Texas at Austin, University of Texas Press

Kurtoglu, T, Campbell, M, Bryant, C, Stone, R and McAdams, D (2005a) *Deriving a component basis for computational functional synthesis* International Conference on Engineering Design, Melbourne, Australia ICED'05

Kurtoglu, T and Campbell, M I (2007) *Assessing the worth of automatically generated design alternatives based on designer preferences* International Conference on Engineering Design, Paris, France ICED'07

Kurtoglu, T, **Campbell, M I**, **Gonzales, J**, **Bryant, C R**, **McAdams, DA**, **Stone, R B** (2005b) Capturing empirically derived design knowledge for creating conceptual design configurations, in *Proceedings of Design Engineering Technical Conference, Sept. 24–28*, Long Beach, California

McAdams, D and Wood, K (2000) Quantitative measures for design by analogy, In: *Proceedings of Design Engineering Technical Conference*, Balitmore, MD

Osborn, A (1957) *Applied imagination* Scribner, New York

Otto, K and Wood, K (2001) *Product design: techniques in reverse engineering and new product development* Prentice-Hall

Otto, K, **Holtta, K** (2004) A multi-criteria framework for screening preliminary product platform concepts, in *Proceedings of Design Engineering Technical Conference*, Salt Lake City, Utah

Pahl, G and Beitz, W (1996) *Engineering design – a systematic approach* (2nd edn) Springer, London

Rohrbach, B (1969) Kreativ nach regeln – methode 635, eine neue technik zum lösen von problemen, *Absatzwirtschaft* Vol 12 pp 73–75

Shah, J J (1998) Experimental investigation of progressive idea generation techniques in engineering design, in *Proceedings of Design Engineering Technical Cconference*, Atlanta, GA

Shah, J J, Kulkarni, S V and Vargas-HernÁndez, N (2000) Evaluation of idea generation methods for conceptual design: effectiveness metrics and design of experiments, *ASME Journal of Mechanical Design* Vol 122 pp 377–384

Shah, J J, Vargas-HernÁndez, N and Smith, S M (2003) Metrics for measuring ideation effectiveness, *Design Studies* Vol 24 pp 111–134

Stone, R and Wood, K (2000) Development of a functional basis for design, *Journal Mechanical Design* Vol 122 pp 359–370

Zwicky, P (1969) *Discovery, invention, research through morphological analysis* McMillan, New York