Ken English
Mem. ASME

Aziz Naim

Kemper Lewis
Professor
Mem. ASME

University at Buffalo,
Buffalo, NY 14260

Susanne Schmidt

Vimal Viswanathan

Julie Linsey
Assistant Professor
Mem. ASME

Daniel A. McAdams
Associate Professor
Mem. ASME

Texas A&M University,
College Station, TX 77843-3123

Bryan Bishop

Matthew I. Campbell
Assistant Professor
Mem. ASME

University of Texas at Austin,
Austin, TX 78712-0292

Kerry Poppa

Robert B. Stone
Professor
Mem. ASME

Oregon State University,
Corvallis, OR 97331

Seth Orsborn
Assistant Professor
Mem. ASME
Bucknell University,
Lewisburg, PA 17837

Impacting Designer Creativity Through IT-Enabled Concept Generation

One of the innovation's fundamental mechanisms, designer creativity, is both unsupported by rigorous information-technology-enabled tools and uncharacterized as a scientific phenomenon. In this paper, we present VISUALIZEIT—a project seeking to identify a scientific basis and develop the supporting cyberinfrastructure needed to facilitate, evaluate, and disseminate information-technology-enabled innovation methodologies that augment designer creativity. This particular research paper describes a method of synthesizing concept representations through the development and expansion of platforms focused on computational concept generation, clustering of design concepts, a repository of archived design knowledge, and an information integration and representation interface. We also present the initial results from implementing VISUALIZEIT using two populations of students. [DOI: 10.1115/1.3484089]

1 Introduction

The redesign of existing products to meet changing customer needs is often an incremental repurposing of previous solutions to reduce both design and manufacturing risk. However, there is significant competitive advantage in the ability to move beyond this incremental change and to enable innovative design. The typical perspective is that innovation often results from the tenuous connection of creative sparks that lead to a new concept. Since these "creative sparks" in conceptual design are intangible, difficult to foster and measure, there is growing interest in formally nurturing creative design approaches by utilizing information technology (IT) tools. While a number of major national reports have called for the revitalization of innovation using new information and knowledge-based tools [1–5], there is a lack of IT-enabled tools to support creativity in conceptual design. Therefore, the impact of such tools on creativity is largely an unknown.

Contributed by the AI/Knowledge Based Systems Committee of ASME for publication in the JOURNAL OF COMPUTING INFORMATION SCIENCE AND ENGINEERING. Manuscript received July 28, 2009; final manuscript received August 2, 2010; published online September 3, 2010. Guest Editors: Ashok K. Goel and Andrés Gómez de Silva Garza.

As technology becomes more diverse, advanced, and global, it becomes difficult for designers to have sufficient resources and expertise to make effective, risk-managed, creative leaps that lead to innovative concepts. This paper describes the research efforts to identify a scientific foundation and to develop the supporting cyberinfrastructure necessary to facilitate, evaluate, and disseminate IT-enabled innovation methodologies that augment designer creativity. This paper presents a novel platform that synthesizes information about computationally generated concepts into an intuitive representation interface for exploration by a designer. In Sec. 2, we review the related work that has established the foundation for our developments. In Sec. 3, we present the design and implementation of the computing platform that we use in our initial creativity experiments, which are presented in Sec. 4 In Secs. 5 and 6, we present observations, conclusions, and directions for future work.

2 Related Work

Our research in information-technology-enabled design innovation leverages fundamental advances in creativity and innovation methods, novel concept generation, visualization techniques, and an empirical approach to measure creativity in design. The relevant work in these core areas is presented in the following sections.

2.1 Creativity in Engineering Design. The initial design phases, including conceptual design, have the most significant impact on product cost [6]. Over 100 formal idea generation techniques exist in areas such as psychology, business, and engineering [7–9]. Techniques include brainstorming developed by Osborn [10] and engineering specific methods (e.g., the theory of inventive problem solving (TRIZ) [11,12]). Group techniques include brainstorming, brainwriting, 6-3-5, C-Sketch, and Gallery [8–10,12–16]. While design catalogs [12] and TRIZ rely on inventories of solutions and principles, virtually all of the other techniques rely on designers' "personal repository" of design knowledge and are thus quite limited.

Formally or informally, designers often reference and base their designs on previous solutions [17–20], known commonly as design-by-analogy. A few formal methods support design-by-analogy, but most rely on a designer's own knowledge. A common finding within controlled analogical reasoning studies is that analogies are difficult to retrieve from memory [21]. However, the mind can be understood as much more than one's memory and arguably includes local environments from which people can extract information [22]. While some postulate that an increase in information-technology-based devices, services, and tools may actually adversely affect our brains [23,24], there are others that believe that expanding digital environments are increasing our mental capabilities [25,26].

By extending a designer's innovative capabilities, the intent is to support one's ability to create novel design ideas. A number of different metrics have been used to evaluate idea generation techniques, including quantity of ideas, number of good ideas, practicality, novelty, and variety [27–33].

2.2 Design Repositories and Concept Generation. Recently, foundations for various portions of concept generation through computational reasoning have been developed. The methods range from late-stage concept generation activities of selecting components to satisfy constraints [34,35] to case-based reasoning that provides the context of suggested components to solve a given function [36,37]. Most methods are based on formalisms for describing the function or purpose in engineering design [38,39] and utilize design databases to allow designers to store and retrieve design knowledge [40–42]. Recently, Oregon State University (OSU) has partnered with University of Texas at Austin (UT) [43,44], Penn State, Virginia Tech, University at Buffalo, Texas A&M University (TAMU), and Bucknell University [45] to expand the types of design information and design tools within such repository.

As empirical knowledge relating components and functions grew and the types of captured information expanded, it led to the development of relational matrices [43] and graph grammar rules [46] based on how components in the repository satisfied functional needs in a particular design. Combined with a search mechanism, the matrices and rules can automatically generate conceptual designs. As a result of the open endedness of conceptual design, a very large number of solutions (i.e., more than a million) can be created. Furthermore, the results showed that subtle challenges in a given design problem were not captured in the specification of initial function, and thus many results were not relevant to the user's needs [47]. A proposed approach is to cluster together designs that have similar attributes and present all of the options in a form that is more manageable by a designer.

Clustering is a general term for a set of exploratory data analysis techniques used to solve grouping or classification problems [48]. Clustering methods tend to be heuristic in nature and are most applicable when there is little information about the underlying structure of the data [49]. The fundamental clustering problem is to sort n observations in d-dimensional space into K groups based on specified similarity criteria [48]. By clustering large numbers of candidate design solutions into groups, more effective representation methods, including visualization tools, can be used to support concept evaluation and exploration.

2.3 Concept Exploration Using Visualization. The advent of virtual reality has provided designers with the capability to see and experience their design before anything is constructed [50–53]. Design visualization has resulted in significant savings in eliminating redesign [54]. Advances in web technologies have expanded the presentation of representations to include web-based visualization of product designs. Current collaborative platforms focus on geometry rather than functionality and therefore limit the creative potential available to the designer. Other approaches, such as graph morphing [55,56], cloud visualization [57], physical programming visualization [58,59], and multidimensional visualization [60], each provide more functional information to the designer in the form of design attribute or optimization information. A "design by shopping" [61] perspective was taken in the effective advanced trade space visualizer (ATSV) system [62] to help guide designers to a solution in an efficient manner. These techniques, which are essentially used as methods of solution validation and subsequent concept selection, become cumbersome for very large problems. Lastly, none of these visualization techniques present information on how well a concept fulfills its intended functions.

3 Design and Implementation

To support designer creativity, a novel clustering approach and a newly developed concept exploration interface are combined with existing distributed information-technology-based resources. The overall VISUALIZEIT platform is presented in Fig. 1. Initially, a designer selects a design problem, and candidate design configurations are automatically generated using a functional model [63], a set of concept generation rules, and information from a digital design repository. The results are stored in a relational database and are grouped into clusters with other concepts that share common design features. The design concepts are then retrieved and presented graphically to the designer. The designer can then graphically explore design alternatives. A more detailed description of each module of the platform follows.

3.1 Design Repository. The design information on existing products used to generate new concepts is integrated, populated, and maintained via a web-based design repository developed by the Design Engineering Lab at OSU [38,39]. The repository database schema establishes what types of design information can be stored, the relationship of those elements, and the extensibility of the database, including new and additional types of design infor-

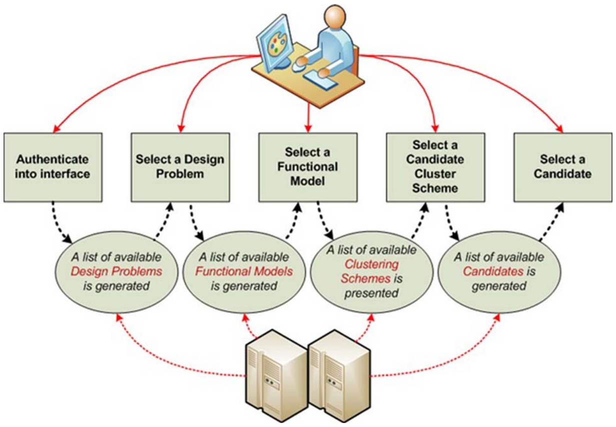

Authenticate into interface
Select a Design Problem
Select a Functional Model
Select a Candidate Cluster Scheme
Select a Candidate
A list of available Design Problems is generated
A list of available Functional Models is generated
A list of available Clustering Schemes is presented
A list of available Candidates is generated

Fig. 1 Overview of VISUALIZEIT approach

mation. Within the OSU design repository, there are two main categories of tables—those that store artifact-specific design data information and those that store taxonomies, or bases, that classify design information. The design repository makes use of several taxonomies to describe information such as functionality, failure modes, manufacturing processes, materials, and color [64].

3.2 Concept Generator. The concept generation method used in this study is based on the driving principle that the design process is a transformation of function to form [65]. The input to the concept generator is a functional model, and the output is one or more component flow graphs (CFGs). A CFG is a graph that shows the connectivity or topology of components in a design, where nodes represent components and arcs represent energy, material, or signal flows between the components. The functional basis [38] terminology is adopted to provide a consistent naming of functions and flows, and component naming follows the convention presented in Ref. [66]. This taxonomy currently includes 136 component types.

Since both input and output types are graphs, the transition from the functional model to CFG is accomplished by a graph rewriting system [67] comprised of 189 graph grammar rules. These rules are created by an empirical process wherein real products are dissected to their individual components, and a functional model is created for each product. In Fig. 2, a hair dryer is shown both as a functional model drawn in FunctionCAD [68] and as a component flow graph. These links between functions and components provide the basis for individual grammar rules.

The grammar rules represent transitions in a state tree [69] where the seed (top of the tree) is the input functional model, and the leaves are completed CFGs. All states between the seed and the leaves are graphs partially comprised of functions and CFGs. The rules follow the generally accepted convention of a left hand side of application conditions (i.e., describing if the rule can be applied and where) and a right hand side of application instructions [70]. Since the left hand side of each rule is a subgraph of a functional model with one or more functions and one or more flows and the right hand side is a subgraph of components and flows, the tree naturally terminates when no rules are applicable on the graph.

For the problem posed in Sec. 4, an initial seed graph functional model produced too many candidate solutions to show a user (i.e., more than a million). Additionally, due to confluence in the rules [71], many of the different paths through the tree arrive at identical solutions. The approach adopted here to present a useful subset to the user is, first, to derive a best sampling of the tree that arrives at the most unique CFGs and then, second, to group these resulting CFGs into clusters. The sampling approach stochastically prefers branches of the search tree that has previously uncalled rules. This sampling method results in approximately a thousand candidates.

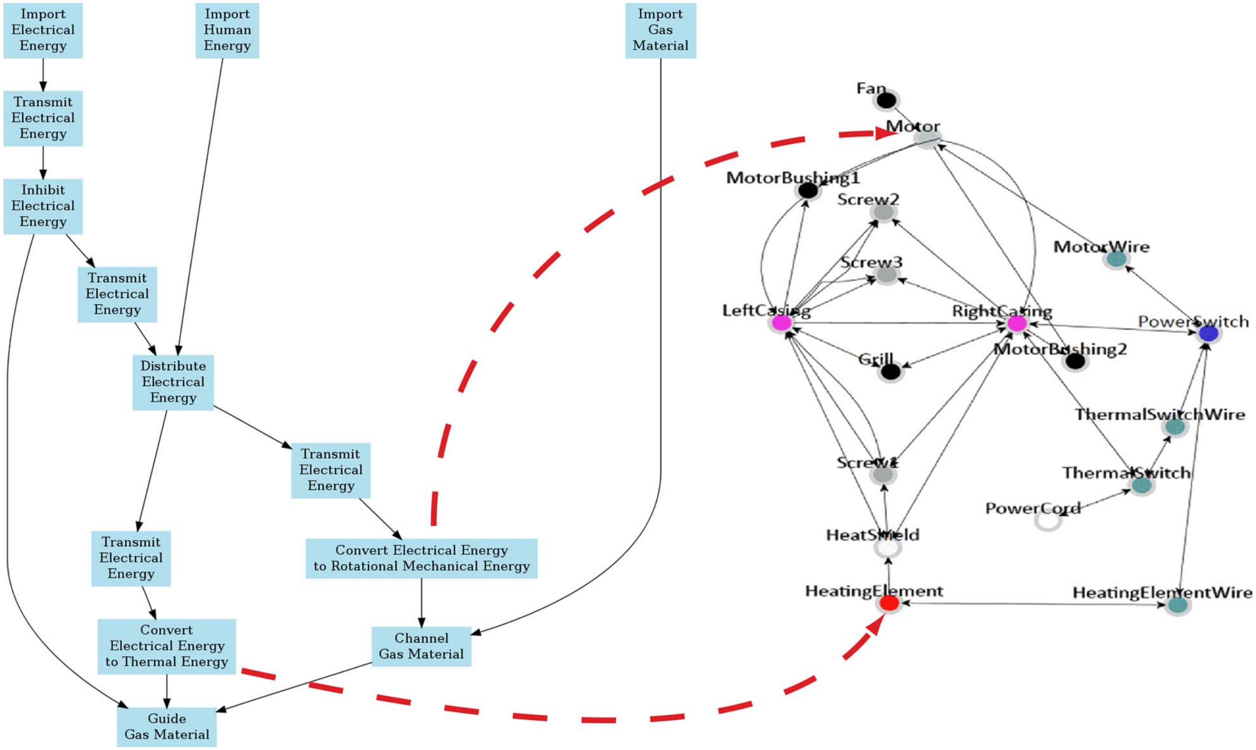


Fig. 2 The functional model (left) and CFG (right) for a hair dryer are shown. Two of the rules that are empirically extracted from the graphs are captured by the dashed arrows.

3.3 Clustering Analysis. The candidates that result from the concept generation phase are clustered into groups of similar concepts before presentation to the user. Based on estimates of working memory [72], seven groups of seven concepts were selected as an initial configuration. To address computational expense resulting from an increase in generated concepts, two different approaches to make the clustering problem more tractable were studied. In the first, the number of concepts is reduced prior to clustering. In the second, the number of dimensions used to describe the concepts is reduced by identifying the principal components before clustering.

Each of the CFGs is converted into a form of a design structure matrix (DSM) [73]. In this application, the DSM is a 138×138 element matrix where the number of elements is determined by the 136 component naming terms [66] plus two generic terms for input and output to capture components that interface with the external environment. Each row and column corresponds to one of the component naming terms. The value of the element at (i,j) is the number of times the ith component is connected to the jth component within a given design. Given that the connection between components is a directed arc, each connection between components is only represented by one value in the matrix, and therefore the matrix is not symmetric. Furthermore, the matrix is very sparse as there are many rows and columns that correspond to components not found in a particular configuration. This matrix enables the comparison of various candidate graphs. Different candidates have different topologies and use different components, but all can be represented by a single sparse matrix following this approach. With these matrices as a common vector space, we are then able to apply standard clustering algorithms.

3.3.1 Complexity Reduction by Euclidean-Norm Method. Given two matrices, $\hat{A}$ and $\hat{B}$ corresponding to two CFGs, we can compute the "distance" between the CFGs by taking the Euclidean norm of the difference of their matrices:

$$\text{distance} = \|\hat{A} - \hat{B}\| \tag{1}$$

From this single value for each pairwise comparison of candidates, we are able to define how different or similar any two CFGs are. These values are then organized into a larger symmetric n-by-n matrix where cell $[i,j]$ (as well as cell $[j,i]$) represents the distance between candidate i and candidate j. To further reduce the space of candidates, the 50 most different CFGs are extracted from the set of 1000. Then, using these distances, the 50 are clustered into seven groups using K-means clustering.

K-means clustering, or Lloyd's algorithm [48], is an iterative clustering algorithm. An evolution of K-means proposed by Arthur and Vassilvitskii is used to partition the data [74]. Initially, a single cluster center is selected at random from the set of candidates, and its distance to each candidate is calculated. Then, a new cluster center is added by selecting another candidate where the chance of selection is proportional the squared distance to the current center. The distance from each candidate to the closest center is recalculated, and a third center is selected. The process is repeated until k cluster centers have been found. Once the initial k cluster centers are found, the standard Lloyd's algorithm is followed. Each candidate is assigned to the cluster with the nearest center. New cluster centers are found by recalculating the cluster centroids. Points are reassigned based on the new cluster centers, and the process continues iteratively until the square error is minimized and the cluster membership stabilizes [48].

3.3.2 Complexity Reduction With Principal Component Analysis. A second approach to the clustering process uses principal component analysis (PCA) to reduce the dimensionality of the problem prior to applying the clustering algorithm [75]. Each 138×138 element matrix is reformulated into a vector by appending each column of the matrix to the preceding column to produce an 18,496 element vector. Note that the input and output elements are removed from consideration for this approach. These vectors are then aggregated into a matrix where columns correspond to candidate CFGs. For the sake of efficiency, rows of the matrix that contain only zeroes are deleted.

PCA is applied by treating each concept as an observation and each component interaction (i.e., row) as a variable. These data are used to generate a new set of variables from linear combinations of the original row variables. All principal components are orthogonal linear combinations of the original variables. Each principal component reproduces a portion of the variation observed in the original model. Initially, the number of principal components equals the number of variables in the original data; variable reduction is achieved by selecting a subset of principal components that adequately mimic the original data. The number of principal components to retain is determined by producing a scree plot of the eigenvalues of the principal components and selecting components from the steepest portion of the curve. The more precise Kaiser criterion, which stipulates that only components with eigenvalues greater than 1 are kept, was also investigated but was found to select too few principal components to adequately reproduce the original data [76].

Once the number of principal components is identified, the original data are transformed onto the principal component space, and a number of clustering techniques can be used to group concepts. In this case, K-means clustering is used as well. After the candidate designs are grouped into clusters, the concepts are presented to the designer using a web-based interface.

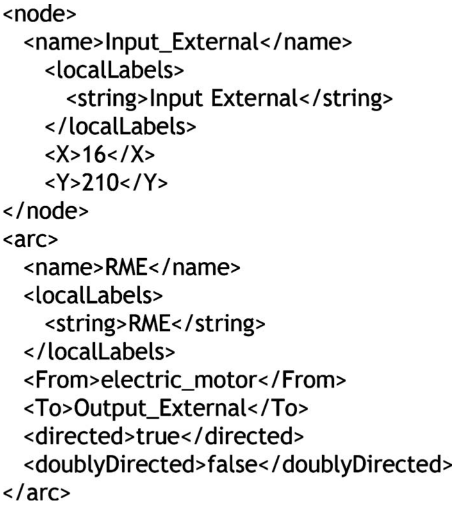
3.4 Web-Based Infrastructure for Concept Exploration. The visualization interface developed here provides designers with graphical representations of concepts to better understand the topology of the generated solutions. This approach synthesizes information, including clustering, connectivity, and functional behavior into a visual format using a combination of web scripting languages (e.g., PHP [77]) and databases. Rather than develop an approach that is tied to specific implementations of computer-based concept generation and archival tools, the architecture described in this section is extensible and can scale with increasing computational power.

The core of the framework is separated into two components, the Data Integration and Validation Module and the Web Server Interface. This separation of data allows for flexibility and extensibility in the development of future clients. The web server interface provides a prototype platform for evaluating the work flow for exploring design alternatives and a flexible platform for initial efficacy trials.

3.4.1 Developing a Common Schema. Efficient functionality of the framework is based on an efficient information flow between the modules. One of the primary issues that must be addressed is the development of a common taxonomy that will be used to pass the information between components of the framework concept generators and design repositories. The advantage of using a consistent communication schema is that no external translation services are needed when different implementations of subcomponents are used, providing a flexible, federated infrastructure.

This work relies on the component naming taxonomy presented in Ref. [66] to describe the system of interest. The schema selected in this work is based on the Extensible Markup Language (XML) [78,79] and was originally developed to capture information from a given functional modeling tool (e.g., GRAPHSYNTH [80]) and pass it to another [81]. Shown in Fig. 3, the schema for each concept is structured using nodes and arcs. Each node entry corresponds to a function present in the overall functional model. Arcs represent the connectivity between the functions and hold information about the type of connection present and its source and sink node.

This approach provides a flexible infrastructure that can accommodate different tools for each component (e.g., concept genera-



```
<node>
  <name>Input_External</name>
    <localLabels>
      <string>Input External</string>
    </localLabels>
    <X>16</X>
    <Y>210</Y>
</node>
<arc>
  <name>RME</name>
  <localLabels>
    <string>RME</string>
  </localLabels>
  <From>electric_motor</From>
  <To>Output_External</To>
  <directed>true</directed>
  <doublyDirected>false</doublyDirected>
</arc>
```

Fig. 3 Example segment of concept description message

tion, clustering, and visualization). The development of a common schema also provides a consistent information flow between the automated concept generator, design repository, and clustering analysis, reducing the likelihood of data loss due to translation between different data formats.

3.4.2 Data Integration and Validation. The architecture developed in this work bridges existing components and translates the data into a meaningful representation to the designer. The process is divided into two stages: data validation and data integration.

Data validation ensures that the framework operates on correct, clean, and useful data. The first stage of data validation is a comparison between data stream and the schema definition document using an XML parser. The second stage of validation ensures consistency with the lexicon developed by Kurtoglu et al. [66]. The third level of validation is a check for completeness. A permissive approach is used, where any data validation problems are highlighted for presentation to the designer because complete designs may not be necessary to stimulate innovative design concepts. Once the information is accepted, the data integration stage parses and integrates the data into a database that establishes the linkage between them. This bridging connects independent sources of information and allows data storage for rapid retrieval by multiple designers.

3.4.3 Development of Concept Exploration Interface. The framework relies on a web server approach to provide the designer with the ability to explore the concept space and gain a better understanding of the topology of the generated solutions by displaying multiple visual representations. The designer is shown a representation of the functional model, grouped candidate solutions, and detailed representations of candidates of interest.

Creating the visual representations integrates information from the automatically generated concepts and the digital design repository data by retrieving archetypal images of a component's type. Transforming the graph theory description of a concept into a human readable representation is a nontrivial task since the placement of nodes and routing of arcs greatly affects human perception. A computer-based graph layout toolkit [82,83] has been incorporated into the representation creation process. This layout tool is used to generate multiple representations of each concept, each using different cues for conveying semantic information for nodes and arcs. A separate algorithm was developed to generate DSM [73] representations of the concept's topology. With the potential for thousands or tens of thousands of design candidates, images are cached on the web server for rapid access to limit the impact of latency on the exploration of the concept space, as discussed in Ref. [84].

The interface is implemented for a sample problem to demonstrate the capabilities of the approach by creating visual representations of a design candidate. The designer is guided in a manner that mimics the progression from a conceptual to a detailed design using the interface. The concept exploration interface shown in Fig. 4 is divided into three distinct regions on the webpage:

- the left menu allowing the designer to make selections
- the top tabs allowing the designer to change between different display formats when available
- the main frame where functional model or conceptual design information (e.g., design structure matrix, component flow graph representation) is displayed.

Before navigating a set of candidate designs, the user first selects a problem of interest in the interface, and a set of possible functional models is provided for review. In Fig. 4, the selection of a *functional model* brings up the preliminary exploration tabs, allowing the visualization of the proposed functional model. A graphical representation of the functional model is automatically generated and is shown with tabs allowing the user to change the presentation of the information between the default graph view and both textual (i.e., matrix of 0's and 1's) and graphical DSM representations.

The default view of the functional model is illustrated in Fig. 5. The automatically generated graph provides basic information about the functions and the flows linking them together. Each graph starts with an input at the top and ends with an output at the bottom. The flows are color and arrow coded to differentiate between flow types. Function and flow names are also provided in order to simplify identification. The user has the ability to choose alternate representations, including a matrix-based or a graphical

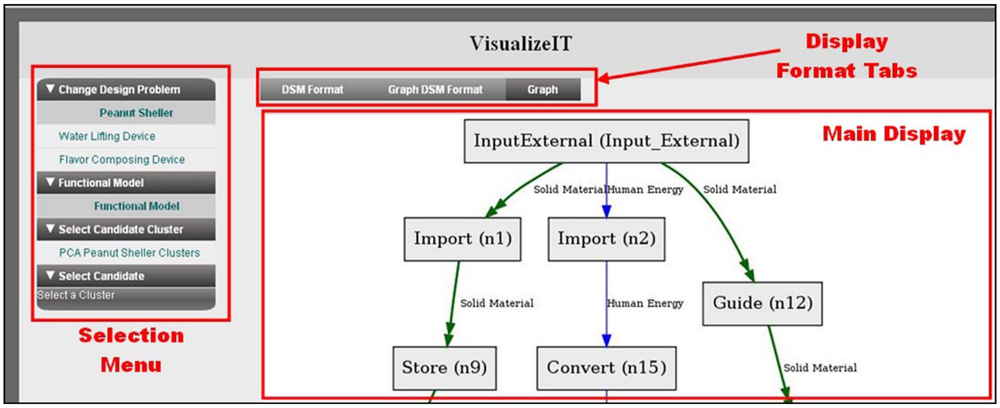


Fig. 4 VISUALIZEIT web interface

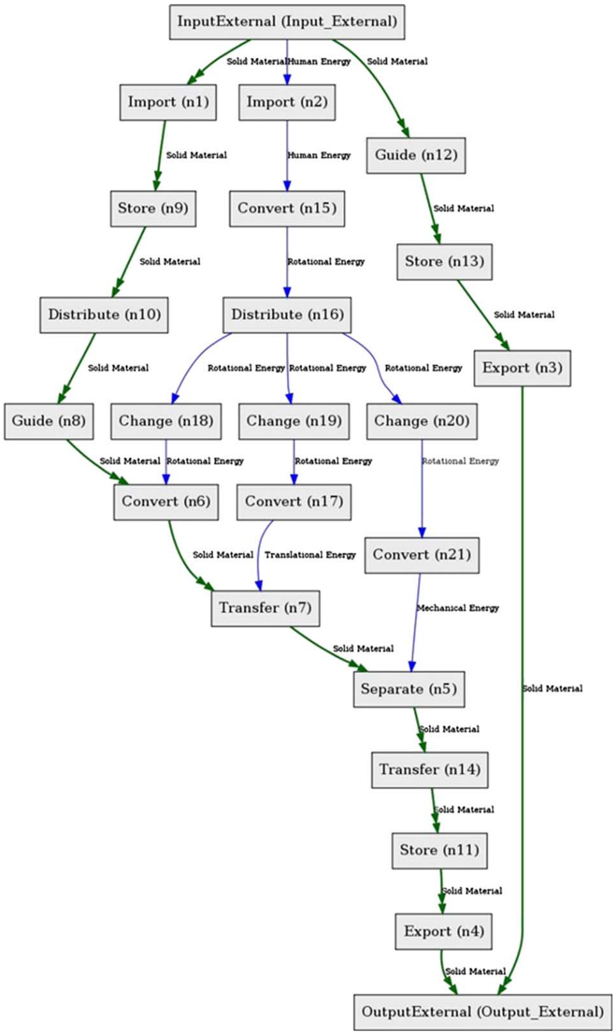


Fig. 5 Functional model of a peanut sheller

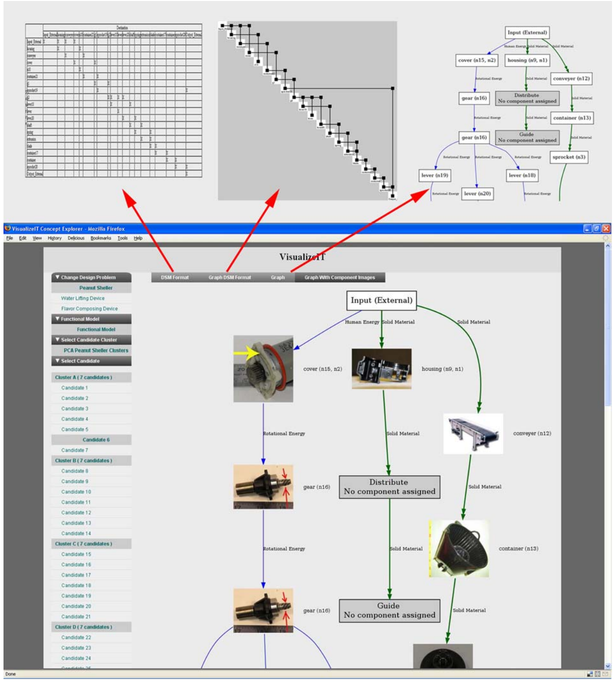


Fig. 6 VISUALIZEIT interface showing a candidate solution (a CFG)

DSM.

After inspecting the functional model, the designer then selects a clustering scheme to explore generated candidates that satisfy the functional model. Each unique candidate group or *cluster* is labeled with a letter, while the included *candidates* are differentiated numerically. The selected problem has seven clusters, each containing approximately seven CFG candidates. For illustration purposes, a single CFG candidate, candidate 6, is used to demonstrate the concept exploration interface in Fig. 6. The figure presents the component flow graph using images of archetypal components. Functions from the original functional model in Fig. 8 that have not been replaced by a component are highlighted. Figure 6 also shows the alternate representations, including textual descriptions of components, a matrix DSM representation, and a graphical DSM representation. In the next section, we discuss the results from a preliminary implementation of the infrastructure.

4 Evaluation

The VISUALIZEIT platform was subjected to a two part evaluation at TAMU and the UT to explore its impact on conceptual design and to provide guidance for future development. The first was a controlled between-subjects experiment comparing idea generation with the VISUALIZEIT tool to a control of participants generating ideas without support. We created an experiment to explore the effects on designer cognition due to the software. The second evaluation seeks user opinions of different approaches to the display and grouping of the CFGs.

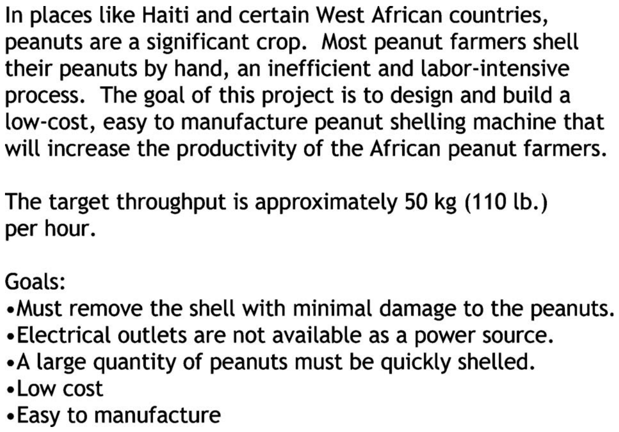
4.1 Experiment 1

4.1.1 Experimental Procedure. The main experimental goal was to provide initial data on how the computer-generated CFGs impact the idea generation process. Therefore, the participants were randomly assigned to two conditions: control (n=15) and software (n=15). In the control group, members were given only the problem description and the functional model (created by the experimenter). In the software condition, participants were also provided the VISUALIZEIT software. To keep the number of displayed CFGs manageable, only 49 CFG layouts were available to the participants through the VISUALIZEIT software.

Two non-native English speaker experimenters, one male and one female, alternately conducted the experiments with each completing about half of each condition at TAMU. For experiments at UT, the software and control were conducted in parallel. Since the experiment was scripted and both experimenters have equal experience conducting the experiment, this was not expected to cause a bias.

In both conditions, participants individually solved the design problems and documented their solutions through sketches and annotations. Two design problems were used, one for training (a water lifting device), and the other for the actual experiment (a peanut sheller, Fig. 7). The design problems were chosen so that senior mechanical engineering students were able to understand them and could create a multitude of solutions, and also so that VISUALIZEIT produced a large set of CFGs.

Members of both conditions started the experiment with a recorded 10 min lecture recapping the key points of functional modeling important for the experiment. The lecture started with defining what a function is, presented black box models showing the overall function of a device, and finally showed a flow-based functional model with the corresponding exploded view of a device. CFGs, similar to the one shown in Fig. 6, were introduced and then briefly explained. Then, by showing a functional model, a CFG, a conceptual sketch, and the sketch of the actual product on the same slide, the connections between these representations were clarified.


In places like Haiti and certain West African countries, peanuts are a significant crop. Most peanut farmers shell their peanuts by hand, an inefficient and labor-intensive process. The goal of this project is to design and build a low-cost, easy to manufacture peanut shelling machine that will increase the productivity of the African peanut farmers.

The target throughput is approximately 50 kg (110 lb.) per hour.

Goals:
- Must remove the shell with minimal damage to the peanuts.
- Electrical outlets are not available as a power source.
- A large quantity of peanuts must be quickly shelled.
- Low cost
- Easy to manufacture

Fig. 7 Peanut sheller design problem

After the lecture, members of both conditions generated ideas with the water lifting device design problem for 20 min. The software group was introduced to the VISUALIZEIT concept explorer prior to starting idea generation and was able to access the CFGs throughout the experiment. The main goal of the training was to allow the software group to familiarize themselves with VISUALIZEIT.

Next, the peanut sheller design problem followed, with 50 min for concept generation. The control group started to generate concepts right away, whereas the software group began the main section with 10 min to compare and contrast the CFGs provided by the VISUALIZEIT interface. After these 10 min, the 50 min concept generation phase started. For both groups, pen colors were exchanged every 10 min, allowing temporal progress to be traced. The experiment concluded with a 3 min questionnaire, measuring demographic data and perceptions of the VISUALIZEIT interface.

4.1.2 Participants. The participants were undergraduate students in their senior year (average $\text{age}=22.4$ with range 21–28), who received extra course credit for participation. The experiment at TAMU was conducted over 4 weeks. Four women and 26 men participated.

4.1.3 Metrics. The quantity of nonredundant ideas developed by each participant was used to compare performance [31,85,86]. The number of unique ideas was defined by the procedure outlined in Linsey et al. [86]. A unique idea is defined as sketched or written entity that solves one or more functions from the functional basis [38]. Inter-rater agreement using Pearson's correlation was 0.9, and the percentage agreement for a random set of ten participants was 93%.

4.1.4 Results. The control group generated statistically more solutions (Fig. 8, t-test quantity: $t=1.8$, $df=28$, and $p=0.08$), and there were no statistically significant differences across the two schools (Fig. 9). The software also seems to focus the participants more on abstract representations of the system, as evident from the diagram in the center of Fig. 10. The participants were clearly able to create concepts based on VISUALIZEIT, which indicates that designers can use the software to augment their process.

The post-experiment survey revealed that participants generally believed that the strictly word-based CFGs (candidates) were not useful, and the picture-based ones received mixed evaluations (Table 1 and Fig. 11). An evaluation of the time spent viewing both types of layouts also indicates that participants spent more time with the picture-based ones. In this experiment, the software condition resulted in significantly fewer ideas, and certain design features such as belts and pulleys were often repeated (Fig. 10). Both results indicate that design fixation occurs [87,88].

Prior literature indicates a number of reasons why CFGs have the potential to cause design fixation [87,89–93]. More ambiguous representations are believed to facilitate creativity [94]. Presenting common solutions rather than unusual solutions also increases the likelihood of design fixation [95,96]. The CFGs generally contain common solutions to particular functions even though the overall layouts can be rather unique. The timing of examples also plays a key role in their adaptation into solutions. If very unique, cross-domain information is presented after a designer has been unable to solve the problem rather than prior to, then the information is much more likely to be implemented [97]. The students had not spent any time attempting to solve the design problem prior to the CFGs being presented; thus they were unlikely to implement any highly unusual solutions.

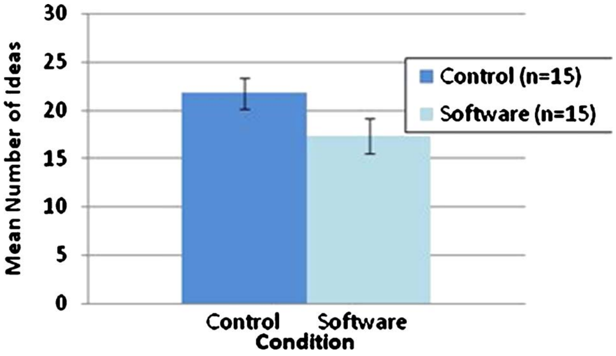


Fig. 8 The number of ideas as a function of condition. Error bars are ±1 one standard error.

4.2 Clustering Experiment. A critical research issue is how the VISUALIZEIT interface should cluster and display sets of concepts. The clustering experiment collected preliminary data on user preferences for the display of the CFGs. The experiment explores three factors related to the display and clustering of the CFGs: (1) the clustering algorithm used to group the CFGs, Euclidean distance, or PCA, (2) the number of CFGs displayed at the same time, and (3) the number of CFGs per cluster for effectively representing the group. The evaluation here explores participant opinions of these factors.

4.2.1 Procedure. The clustering experiment was run after the evaluation of the software. If participants were in the software group, they were already familiar with the CFGs and the clustering experiment immediately followed. If the participants were in the control group, they were introduced to VISUALIZEIT and then generated ideas on the peanut sheller problem for 15 min prior to starting the clustering experiment. In the first stage of the experiment, the participants were given three sets of CFGs, one set with PCA clusters, another with Euclidean-distance clusters, and a third with random sets of CFGs from the other two clusters. Each set contained seven clusters of five CFGs. Participants were asked to

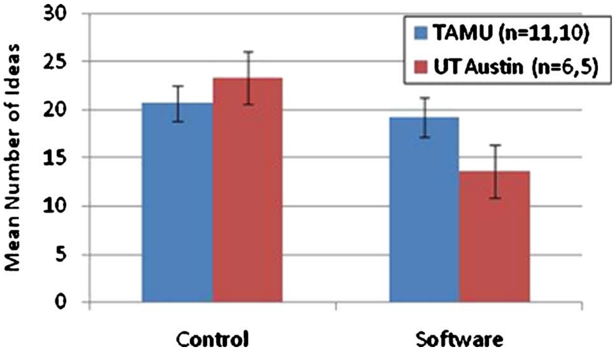
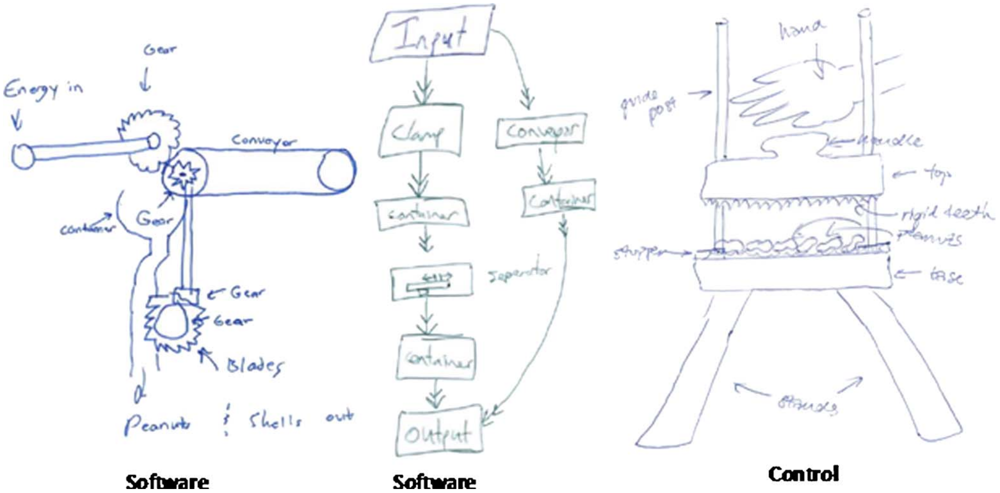


Fig. 9 Mean number of ideas as a function of condition and location. Error bars are ±1 standard error.



Software Software Control

Fig. 10 Some example results. The VISUALIZEIT software tended to focus participants on more abstract system representations (center).

review the sets and then rank them from most to least useful.

In the next stage, the participants were given sheets containing one, three, or five CFGs from the same cluster and were asked their preferences. In the final stage, the participants were given the same CFGs arranged in two different formats. In the first format, five CFGs were given side by side, whereas in the second format, they were given a stack of paper stapled together. The participants were then asked which representations they prefer. The experiment ended with a short survey.

4.2.2 Results of the Clustering Experiment. The clustering experiment provides some preliminary guidance on how the candidates should be displayed. Participants had no preference for a particular clustering approach (number of participants who preferred each PCA=4, Euclidean distance=3, random=5, all equal=12, and no response=1). They did prefer to see three representative candidates for each cluster rather than only one or five (number of participants: one candidate=3, three candidates=14, five candidates=8, and no response=1). In addition, almost no one wanted more than five shown (more than five candidates per cluster: yes=2, no=22, and no response=1). In general, they preferred to see multiple candidates from the same cluster side by side rather than having only one visible, as is currently implemented (16=side-by-side and 9=one visible).

5 Conclusions

In this work, we have developed a computational platform to support design innovation in the conceptual design of products. Specifically, VISUALIZEIT provides concept variant clustering approaches while offering concept variant visualization capabilities, allowing designers to explore innovative designs.

Developing the platform required a substantial integration of the repository, concept generation, visualization, and clustering tools. The integration allowed for a rigorous study of the VISUALIZEIT platform and its impact on student creativity. While the number of concepts for the VISUALIZEIT group was fewer than the control group, student fixation on the resulting concepts is a likely cause. Preliminary results also seemed to indicate that the graphical representations allowed for more insights than standard representations did. Also, since participants did not prefer either Euclidean distance or PCA over random clustering, more study on alternative clustering approaches is necessary.

VISUALIZEIT provides a rich foundation for the further assessment of innovation support tools by testing the effect of the digital methods and capabilities on the creative output of designers. The assessment protocol will focus on measuring the impact of the generation, clustering, filtering, and visualization methods, leading to an understanding of how previous design knowledge can best be presented to increase the creative output of designers.

6 Future Work

With regard to concept generation and clustering to support future VISUALIZEIT development, investigation of approaches to guide concept generation instead of relying on a clustered set from all possible concept variants is needed. In this scenario, a small number of randomly generated concept variants could be computed, clustered, and then presented to the designer for feedback. Favorable concept clusters could then be used to guide the selection of grammar rules to be applied to produce additional concept variants, which would again be clustered and presented to the designer. This process could continue until the designer is satisfied with the concept variant results.

Future experimental work includes experiments done with novices and expert designers. Significant research will need to directly measure the effects on the idea generation of the different approaches to displaying the candidates to the participants. Participants may prefer one approach (e.g., viewing candidates one at

Table 1 Participant survey results evaluating the CFGs (candidates)

Survey question	MEAN (Std. Dev.) (Strongly disagree=1, disagree=2, neither=3, agree=4, STRONGLY agree=5)
The word-based CFG representation helped me generate ideas.	2.3 (1.0) Disagree
The picture-based CFG representation (the candidates under the graph with component images tab) helped me generate ideas.	3.2 (1.4) Neither agree nor disagree

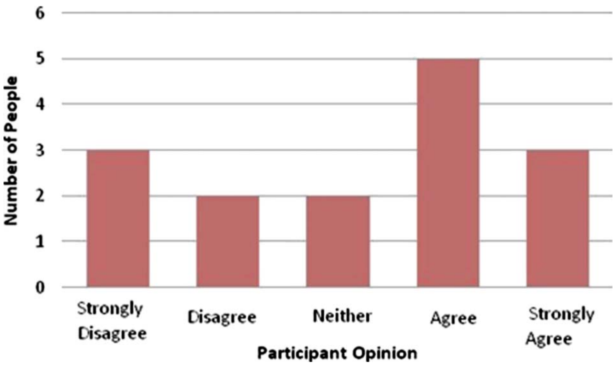


Fig. 11 Participant opinion on if the picture-based CFGs assisted with generating ideas

a time), but a different approach (e.g., viewing multiple candidates side by side) may actually enhance their idea generation more. Research also needs to evaluate what types of CFGs augment the designer best. For example, it may be best to present the user with only a few clusters of very unusual layouts, or the representation of the component may be critical. Each of the components could be represented in a CFG with words, pictures, sketches, or computer aided design (CAD) models. The representation of the components also needs to be explored in terms of causing designer fixation. Common solutions are known to cause designer fixation, and ambiguous representations such as sketches are believed to enhance creativity, so sketched components may be better than pictures of components. Future experiments need to continue to evaluate designers augmented with the software in comparison to designers without software support.

A major limitation for demonstrating the potential of the software is that database is a relatively limited design space consisting of only about 200 products. Strictly for demonstrating the potential of the VISUALIZEIT software, a future experiment needs to prepopulate the database with a large variety of solutions specific to the problem under consideration. For example, for the peanut sheller problem, the database contains few solutions for removing the shell relative to the vast known solutions for this function. A future experiment will prepopulate the database with a large range of both common and unusual solutions for the functions within the peanut sheller. This would provide a better evaluation of the potential for the software.

Acknowledgment

We are grateful to the National Science Foundation (Grant Nos. IIS-0841379—Missouri S&T, IIS-0840969—Texas A&M, IIS-0841389—Univ. at Buffalo, SUNY, IIS-0848244—U. of Texas-Austin), and the New York State Office of Science, Technology and Academic Research (Grant No. M050100) for their support of this work.

References

[1] Augustine, N. R., 2007, “Is America Falling Off the Flat Earth?,” Rising Above the Gathering Storm Committee, National Academy of Sciences, National Academy of Engineering, and Institute of Medicine of the National Academies.

[2] National Innovation Initiative Summit, 2004, “Innovate America: National Innovation Initiative Summit and Report: Thriving in a World of Challenge and Change,” Council on Competitiveness, Report.

[3] Popkin, J., and Kobe, K., 2006, “U.S. Manufacturing Innovation at Risk,” Council of Manufacturing Associations and the Manufacturing Institute, Washington, D.C.

[4] President’s Council of Advisors on Science and Technology, 2004, “Sustaining the Nation’s Innovation Ecosystems: Information Technology Manufacturing and Competitiveness,” White House Office of Science and Technology Policy, Washington, DC.

[5] Task Force on the Future of American Innovation, 2005, “The Knowledge Economy Is the United States Losing Its Competitive Edge? Benchmarks for Our Innovation Future,” Task Force on the Future of American Innovation, Washington, D.C.

[6] Römer, A., Pache, M., Weißhahn, G., Lindemann, U., and Hacker, W., 2001, “Effort-Saving Product Representations in Design—Results of a Questionnaire Survey,” Des. Stud., **22**(6), pp. 473–490.

[7] Adams, J. L., 1986, *Conceptual Blockbusting: A Guide to Better Ideas*, Addison-Wesley, Reading, MA.

[8] Higgins, J. M., 1994, *101 Creative Problem Solving Techniques: The Handbook of New Ideas for Business*, New Management, Winter Park, FL.

[9] Van Gundy, A. B., 1998, *Techniques of Structured Problem Solving*, Van Nostrand Reinhold, New York.

[10] Osborn, A. F., 1963, *Applied Imagination; Principles and Procedures of Creative Problem-Solving*, Scribner, New York.

[11] Altshuller, G. S., 1984, *Creativity as an Exact Science: The Theory of the Solution of Inventive Problems*, Gordon and Breach, New York.

[12] Pahl, G., and Beitz, W., 1996, *Engineering Design: A Systematic Approach*, Springer, New York.

[13] Otto, K. N., and Wood, K. L., 2001, *Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development*, Prentice-Hall, Upper Saddle River, NJ.

[14] Paulus, P. B., and Yang, H. C., 2000, “Idea Generation in Groups: A Basis for Creativity in Organizations,” Org. Behav. Hum. Decis. Process, **82**(1), pp. 76–87.

[15] Shah, J. J., Vargas-Hernandez, N., Summers, J. D., and Kulkarni, S., 2001, “Collaborative Sketching (C-Sketch)—An Idea Generation Technique for Engineering Design,” J. Creat. Behav., **35**(3), pp. 168–198.

[16] van der Lugt, R., 2002, “Brainsketching and How It Differs From Brainstorming,” Creat. Innov. Manag., **11**(1), pp. 43–54.

[17] Casakin, H., and Goldschmidt, G., 1999, “Expertise and the Use of Visual Analogy: Implications for Design Education,” Des. Stud., **20**(2), pp. 153–175.

[18] Christensen, B. T., and Schunn, C. D., 2007, “The Relationship of Analogical Distance to Analogical Function and Pre-Inventive Structure: The Case of Engineering Design,” Mem. Cognit., **35**(1), pp. 29–38.

[19] Eckert, C. M., and Stacey, M., 2005, “Reasoning by Reference to Past Designs,” Proceedings of Studying Designers ‘05, Key Centre of Design Computing and Cognition, University of Sydney, J. S. Gero, ed.

[20] Leclercq, P., and Heyligen, A., 2002, “5,8 Analogies Per Hour, a Designer’s View on Analogical Reasoning,” Proceedings of AID’02, Seventh International Conference on Artificial Intelligence in Design, Cambridge, UK.

[21] Gick, M. L., and Holyoak, K. J., 1980, “Analogical Problem Solving,” Cognit Psychol., **12**(3), pp. 306–355.

[22] Clark, A., and Chalmers, D. J., 1998, “The Extended Mind,” Analysis, **58**(1), pp. 7–19.

[23] Carr, N., 2008, “Is Google Making Us Stupid? Why You Can’t Read the Way You Used To,” The Atlantic Monthly, **302**(1), pp. 56–63.

[24] Wolf, M., and Stoodley, C. J., 2008, *Proust and the Squid: The Story and Science of the Reading Brain*, Icon, Thriplow.

[25] Crystal, D., 2008, *Txtng: The Gr8 Db8*, Oxford University Press, New York.

[26] Zimmer, C., 2009, “How Google Is Making Us Smarter,” Discover Magazine, **30**(2), pp. 30–32.

[27] Bouchard, T. J., Jr., 1969, “Personality, Problem-Solving Procedure, and Performance in Small Groups,” J. Appl. Psychol., **53**(1), pp. 1–29.

[28] Mullen, B., Johnson, C., and Salas, E., 1991, “Productivity Loss in Brainstorming Groups: A Meta-Analytic Integration,” Basic Applied Social Psychology, **12**(1), pp. 3–23.

[29] Parnes, S. J., and Meadow, A., 1959, “Effects of ‘Brainstorming’ Instructions on Creative Problem Solving by Trained and Untrained Subjects,” J. Educ. Psychol., **50**(4), pp. 171–176.

[30] Shah, J. J., Kulkarni, S. V., and Vargas-Hernandez, N., 2000, “Evaluation of Idea Generation Methods for Conceptual Design: Effectiveness Metrics and Design of Experiments,” ASME J. Mech. Des., **122**(4), pp. 377–384.

[31] Shah, J. J., Smith, S. M., and Vargas-Hernandez, N., 2003, “Metrics for Measuring Ideation Effectiveness,” Des. Stud., **24**(2), pp. 111–134.

[32] Torrance, E. P., 1962, *Guiding Creative Talent*, Prentice-Hall, Englewood Cliffs, NJ.

[33] Van Der Lugt, R., 2005, “How Sketching Can Affect the Idea Generation Process in Design Group Meetings,” Des. Stud., **26**(2), pp. 101–122.

[34] Bracewell, R., and Johnson, A. L., 1999, “From Embodiment Generation to Virtual Prototyping,” Proceedings of the International Conference on Engineering Design, ICED 99, Munich, Germany, Vol. 2, pp. 685–690.

[35] Thornton, A. C., and Johnson, A. L., 1996, “Cadet: A Software Support Tool for Constraint Processes in Embodiment Design,” Res. Eng. Des., **8**(1), pp. 1–13.

[36] Goel, A. K., De Silva Garza, A. G., Grue, N., Murdock, J. W., Recker, M., and Govindaraj, T., 1996, *Lecture Notes in Computer Science*, Springer, Berlin, p. 493.

[37] Goel, A. K., and Murdock, J. W., 1996, “Meta-Cases: Explaining Case-Based Reasoning,” Proceedings of the Advances in Case-Based Reasoning: Third European Workshop, EWCBR-96, Lausanne, Switzerland, I. Smith and B. Faltings, eds., pp. 150–163.

[38] Hirtz, J., Stone, R. B., Mcadams, D. A., Szykman, S., and Wood, K. L., 2001, “A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts,” Res. Eng. Des., **13**(2), p. 65.

[39] Stone, R. B., and Wood, K. L., 2000, “Development of a Functional Basis for Design,” ASME J. Mech. Des., **122**(4), pp. 359–370.

[40] Xue, D., Yang, H., and Tu, Y. L., 2006, “Modeling of Evolutionary Design Database,” ASME J. Comput. Inf. Sci. Eng., **6**(1), pp. 22–32.

[41] Rachuri, S., Han, Y.-H., Foufou, S., Feng, S. C., Roy, U., Wang, F., Sriram, R. D., and Lyons, K. W., 2006, "A Model for Capturing Product Assembly Information," ASME J. Comput. Inf. Sci. Eng., **6**(1), pp. 11–21.

[42] Bohm, M. R., Stone, R. B., and Szykman, S., 2005, "Enhancing Virtual Product Representations for Advanced Design Repository Systems," ASME J. Comput. Inf. Sci. Eng., **5**(4), pp. 360–372.

[43] Bryant, C., Mcadams, D., Stone, R., Kurtoglu, T., and Campbell, M. I., 2005, "A Computational Technique for Concept Generation," Proceedings of the ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Long Beach, CA, Paper No. DETC2005/DTM-85323.

[44] Bryant, C., Stone, R., Mcadams, D., Kurtoglu, T., and Campbell, M., 2005, "Concept Generation From the Functional Basis of Design," Proceedings of the International Conference on Engineering Design, ICED '05, Melbourne, Australia.

[45] Shooter, S. B., Simpson, T. W., Kumaray, S. R. T., Stone, R. B., and Terpenny, J. P., 2005, "Toward a Multi-Agent Information Infrastructure for Product Family Planning and Mass Customization," International Journal of Mass Customization, **1**(1), pp. 134–155.

[46] Kurtoglu, T., and Campbell, M. I., 2009, "Automated Synthesis of Electromechanical Design Configurations From Empirical Analysis of Function to Form Mapping," J. Eng. Design, **20**(1), pp. 83–104.

[47] Bohm, M., 2009, "Information Archival and Reuse: Drawing Conclusions From the Past," Ph.D. thesis, Missouri University of Science and Technology, Rolla, Missouri.

[48] Backer, E., 1995, *Computer-Assisted Reasoning in Cluster Analysis*, Prentice-Hall, New York.

[49] Anderberg, M. R., 1982, *Cluster Analysis for Applications*, Academic, New York.

[50] Ames, A. L., Nadeau, D. R., and Moreland, J. L., 1997, *The VRML 2.0 Sourcebook*, Wiley, New York.

[51] Bryson, S., and Levit, C., 1992, "The Virtual Wind Tunnel," IEEE Comput. Graphics Appl., **12**(4), pp. 25–34.

[52] Cruz-Neira, C., Sandin, D. J., and Defanti, T. A., 1993, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the Cave," Comput. Graphics, **27**, pp. 135–142.

[53] Geist, G. A., Kohl, J. A., and Papadopoulos, P. M., 1997, "Cumulvs: Providing Fault Tolerance, Visualization, and Steering of Parallel Applications," The International Journal of High Performance Computing Applications, **11**(3), pp. 224–235.

[54] Price, A. M., 1998, "Virtual Product Development 'Case Study of the T-45 Horizontal Stabilator'," Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Long Beach, CA, Paper No. A98-25247.

[55] Winer, E. H., and Bloebaum, C. L., 2002, "Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization. Part I: Method Development," Struct. Multidiscip. Optim., **23**(6), pp. 412–424.

[56] Winer, E. H., and Bloebaum, C. L., 2002, "Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization. Part II: Method Validation," Struct. Multidiscip. Optim., **23**(6), pp. 425–435.

[57] Eddy, J., and Lewis, K., 2002, "Visualization of Multi-Dimensional Design and Optimization Data Using Cloud Visualization," Proceedings of 2002 ASME Design Technical Conferences, Design Automation Conference, Montreal, Canada, Paper No. DETC02/DAC-02006.

[58] Mattson, C., Mullur, A., and Messac, A., 2004, "Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space," Eng. Optimiz., **36**(6), pp. 721–740.

[59] Mattson, C. A., and Messac, A., 2003, "Concept Selection Using S-Pareto Frontiers," AIAA J., **41**(6), pp. 1190–1198.

[60] Ribarsky, W., Ayers, E., Eble, J., and Mukherjea, S., 1994, "Glyphmaker: Creating Customized Visualizations of Complex Data," IEEE Trans. Comput., **27**(7), pp. 57–64.

[61] Balling, R., 1999, "Design by Shopping: A New Paradigm," Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization, Buffalo, NY, pp. 295–297.

[62] Stump, G., Lego, S., Yukish, M., Simpson, T. W., and Donndelinger, J. A., 2009, "Visual Steering Commands for Trade Space Exploration: User-Guided Sampling With Example," ASME J. Comput. Inf. Sci. Eng., **9**(4), p. 044501.

[63] Hoyle, C., Tumer, I. Y., Mehr, A. F., and Chen, W., 2009, "Health Management Allocation During Conceptual System Design," ASME J. Comput. Inf. Sci. Eng., **9**(2), p. 021002.

[64] Bohm, M. R., Stone, R. B., Simpson, T. W., and Steva, E. D., 2008, "Introduction of a Data Schema to Support a Design Repository," Comput.-Aided Des., **40**(7), pp. 801–811.

[65] Kurtoglu, T., Campbell, M. I., Gonzalez, J., Bryant, C. R., Stone, R. B., and Mcadams, D. A., 2005, "Capturing Empirically Derived Design Knowledge for Creating Conceptual Design Configurations," Proceedings of the 2005 ASME International Design Theory and Methodology Conference, Long Beach, CA, Paper No. IDETC/DTM-84405, pp. 249–257.

[66] Kurtoglu, T., Campbell, M. I., Arnold, C. B., Stone, R. B., and Mcadams, D. A., 2009, "A Component Taxonomy as a Framework for Computational Design Synthesis," ASME J. Comput. Inf. Sci. Eng., **9**(1), p. 011007.

[67] Nagl, M., 1983, *Bibliography on Graph Rewriting Systems (Graph-Grammars)*, Fachbereich Mathematik der Universität Osnabrück, Osnabrück.

[68] Nagel, R. L., Perry, K. L., Stone, R. B., and Mcadams, D. A., 2009, "Functioncad: A Functional Modeling Application Based on the Function Design Framework," Proceedings of the 2009 ASME Design Engineering Technical Conferences and Computers and Information in Engineering, San Diego, CA, Paper No. DETC2009- 87010.

[69] Nilsson, N. J., 1982, *Principles of Artificial Intelligence*, Springer, Berlin.

[70] Ehrig, H., 2006, *Fundamentals of Algebraic Graph Transformation*, Springer, Berlin.

[71] Heckel, R., Kuster, J. M., and Taentzer, G., 2002, "Confluence of Typed Attributed Graph Transformation Systems," Lect. Notes Comput. Sci., **2505**, pp. 161–176.

[72] Miller, G. A., 1956, "The Magical Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychol. Rev., **63**(2), pp. 81–97.

[73] Steward, D. V., 1981, "The Design Structure System: A Method for Managing the Design of Complex Systems," IEEE Trans. Eng. Manage., **28**(3), pp. 71–74.

[74] Arthur, D., and Vassilvitskii, S., 2007, "K-Means++: The Advantages of Careful Seeding," Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035.

[75] Orsborn, S., Boatwright, P., and Cagan, J., 2008, "Identifying Product Shape Relationships Using Principal Component Analysis," Res. Eng. Des., **18**(4), pp. 163–180.

[76] Kaiser, H. F., 1960, "The Application of Electronic Computers to Factor Analysis," Educ. Psychol. Meas., **20**(1), pp. 141–151.

[77] Lerdorf, R., and Tatroe, K., 2002, *Programming Php*, O'Reilly, Sebastopol, CA/Beijing.

[78] Extensible Markup Language (Xml), http://www.w3.org/XML/

[79] Birbeck, M., Duckett, J., Gudmundsson, O., Kobak, P., Lenz, E., Livingstone, S., Marcus, D., Mohr, S., Pinnock, J., Visco, K., Watt, A., Williams, K., Zaev, Z., and Ozu, N., 2001, *Professional XML (Programmer to Programmer)*, 2nd ed., Wrox Press, Birmingham, UK.

[80] GRAPHSYNTH, software for generative grammars and creative search, http://graphsynth.com/

[81] Szykman, S., Racz, J. W., and Sriram, R. D., 1999, "The Representation of Function in Computer-Based Design," Proceedings of the ASME Design Engineering Technical Conferences (11th International Conference on Design Theory and Methodology, Las Vegas, NV, Paper No. DETC99/DTM-8742.

[82] Ellson, J., Gansner, E., Koutsofios, E., North, S. C., and Woodhull, G., 2004, "Graphviz and Dynagraph—Static and Dynamic Graph Drawing Tools," *Graph Drawing Software*, M. Junger and P. Mutzel, eds., Springer-Verlag, New York, pp. 127–148.

[83] Gansner, E. R., and North, S. C., 2000, "An Open Graph Visualization System and Its Applications to Software Engineering," Softw.: Pract. Exp., **30**(11), pp. 1203–1233.

[84] Simpson, T., Barron, K., Rothrock, L., Frecker, M., Barton, R., and Ligetti, C., 2007, "Impact of Response Delay and Training on User Performance With Text-Based and Graphical User Interfaces for Engineering Design," Res. Eng. Des., **18**(2), pp. 49–65.

[85] Kurtoglu, T., Campbell, M., and Linsey, J., 2009, "An Experimental Study on the Effects of a Computational Design Tool on Concept Generation," Des. Stud., **30**(6), pp. 676–703.

[86] Linsey, J., Green, M. G., Murphy, J. T., Wood, K. L., and Markman, A. B., 2005, "Collaborating to Success: An Experimental Study of Group Idea Generation Techniques," Proceedings of the ASME Design Theory and Methodology Conference, Long Beach, CA.

[87] Jansson, D. G., and Smith, S. M., 1991, "Design Fixation," Des. Stud., **12**(1), pp. 3–11.

[88] Linsey, J., Tseng, I., Fu, K., Cagan, J., Wood, K., and Schunn, C., 2010, "A Study of Design Fixation, Its Mitigation and Perception in Engineering Design Faculty," ASME J. Mech. Des., **132**(4), p. 041003.

[89] Jaarsveld, S., and Van Leeuwen, C., 2005, "Sketches From a Design Process: Creative Cognition Inferred From Intermediate Products," Cogn. Sci., **29**(1), pp. 79–101.

[90] Marsh, R. L., Ward, T. B., and Landau, J. D., 1999, "The Inadvertent Use of Prior Knowledge in a Generative Cognitive Task," Mem. Cognit., **27**(1), pp. 94–105.

[91] Purcell, A. T., and Gero, J. S., 1996, "Design and Other Types of Fixation," Des. Stud., **17**(4), pp. 363–383.

[92] Smith, S. M., and Blankenship, S. E., 1991, "Incubation and the Persistence of Fixation in Problem Solving," Am. J. Psychol., **104**(1), pp. 61–87.

[93] Ward, T. B., 1994, "Structured Imagination: The Role of Category Structure in Exemplar Generation," Cognit Psychol., **27**(1), pp. 1–40.

[94] Shah, J. J., 1998, "Experimental Investigation of Progressive Idea Generation Techniques in Engineering Design," Proceedings of the ASME Design Engineering Technical Conferences, Atlanta, GA, Paper No. DETC98/DTM-5676.

[95] Dugosh, K. L., and Paulus, P. B., 2005, "Cognitive and Social Comparison Processes in Brainstorming," J. Exp. Soc. Psychol., **41**(3), pp. 313–320.

[96] Perttula, M., and Sipila, P., 2007, "The Idea Exposure Paradigm in Design Idea Generation," J. Eng. Design, **18**(1), pp. 93–102.

[97] Tseng, I., Moss, J., Cagan, J., and Kotovsky, K., 2008, "The Role of Timing and Analogical Similarity in the Stimulation of Idea Generation in Design," Des. Stud., **29**(3), pp. 203–221.