*Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (2017), *31*, 501–511.

doi:10.1017/S0890060417000488

# Transforming functional models to critical chain models via expert knowledge and automatic parsing rules for design analogy identification

MALENA AGYEMANG,[1] JULIE LINSEY,[2] AND CAMERON J. TURNER[1]

[1]Design Innovation and Computational Engineering Laboratory, Clemson University, Clemson, South Carolina, USA
[2]Innovation, Design Reasoning, Engineering Education, and Methods Lab, Georgia Institute of Technology, Atlanta, Georgia, USA

(Received November 11, 2016; Accepted May 25, 2017)

**Abstract**

Critical chains composed of critical flows and functions have been demonstrated as an effective qualitative analogy retrieval approach based on performance metrics. In prior work, engineers used expert knowledge to transform functional models into critical chain models, which are abstractions of the functional model. Automating this transformation process is highly desirable so as to provide for a robust transformation method. Within this paper, two paradigms for functional modeling abstraction are compared. A series of pruning rules provide an automated transformation approach, and this is compared to the results generated previously through an expert knowledge approach. These two approaches are evaluated against a set of published functional models. The similarity of the resulting transformation of the functional models into critical chain models is evaluated using a functional chain similarity metric, developed in previous work. Once critical chain models are identified, additional model evaluation criteria are used to evaluate the utility of the critical chain models for design analogy identification. Since the functional vocabulary acts as a common language among designers and engineers to abstract and represent critical design artifact information, analogous matching can be made about the functional vocabulary. Thus, the transformation of functional models into critical chain models enables engineers to use functional abstraction as a mechanism to identify design analogies. The critical flow rule is the most effective first step when automatically transforming a functional model to a critical chain model. Further research into more complex critical chain model architectures and the interactions between criteria is merited.

**Keywords:** Analogy Matching; Automated Abstraction; Design by Analogy; Functional Criteria Metrics; Functional Modeling

## 1. INTRODUCTION

Often, multiple iterations of abstraction and deabstraction are necessary to yield a design solution. The use of analogies by experienced designers is common, while the lack of expertise by novice designers often limits their ability to identify and apply analogies to the design problem. However, computational approaches to assist both novice and experienced designers explore design analogies are of increasing interest to the engineering design community. In order for a tool to computationally match analogies based on abstraction, an automated abstracted design model is needed to support a comparative method of analogy identification. In this paper, we examine the validity of an abstraction approach that transforms a functional model into a critical chain model. With this critical chain model, a combination of similarity and architectural criteria can then be employed to identify design analogies. This analogy identification approach has been demonstrated using expert knowledge to transform functional models to critical chain models and employing a single criterion matching approach. However, applying expert knowledge to transform functional models into critical chain models was time intensive, and the single criterion matching approach provided clear indications the matching was a multiple-criterion problem. This paper evaluates a set of pruning rules as potential candidates for the automated transformation of functional models into critical chain models and a set of criteria for their effectiveness in identifying known analogies.

Reprint requests to: Cameron J. Turner, Design Innovation and Computational Engineering Laboratory, Fluor Daniel Engineering Innovation Building, Clemson University, Clemson, SC 29634, USA. E-mail: cturne9@clemson.edu

## 2. FUNCTIONAL MODELING

A common abstraction tool used by design engineers is the creation of a functional model (Qian & Gero, 1996; Hirtz et al., 2002; Pahl et al., 2007). This paper leverages a functional modeling technique that consists of flows representing the energy, materials, and signals acting upon and within the functional model, which act upon the flows (Otto & Wood, 2000; Hirtz et al., 2002; Pahl et al., 2007; Dieter & Smith, 2009), as shown in Figure 1.

Both the functions and the flows are described using a limited vocabulary of terms defined in the revised functional basis (Hirtz et al., 2002; Nagel & Bohm, 2011). The flows of a functional model obey the laws of energy and mass conservation. The resulting network of flows and functions forms a graph-based model that abstracts the function(s) of a system from the form of the system. The key benefit of a functional model abstraction to the design engineer is the separation of *function*, what must be done, and from *form*, how it is done.

### 2.1. Function modeling for design analogies

Studies focused on the activities of designers indicate that previous experience is often used to identify solutions (Casakin & Goldschmidt, 1999; Ball et al., 2004; Christensen & Schunn, 2007; Chan & Schunn, 2014), and those solutions are implemented into new design through analogy. The cognitive mechanism of analogical reasoning is applied through the process of abstracting and deabstracting the design. Established analogy tools do exist, but many of these systems generate analogies via a verbal problem abstraction and perform matches through linguistic similarity and keyword searches (Chakrabarti et al., 2005; Nagel & Bohm, 2011; Vattam et al., 2011; Linsey et al., 2012; Goel et al., 2013; see also Biomimicry 3.8 Institute at http://www.asknature.org; Biomimicry Group at http://biomimicry.net; and Biomimicry Institute at http://www.asknature.org/article/view/why_asknature). Functional model abstractions also are used to estimate market-based price prediction models, product assembly time, and manufacturing costs (Caldwell & Mocko, 2008; Mathieson et al., 2013; Namouz & Summers, 2013, 2014; Owensby & Summers, 2014; Summers et al., 2014; Gill et al., 2016).

The Design-Analogy Performance Parameter System (D-APPS) is a tool that computationally identifies design analogies (Lucero, 2014; Lucero et al., 2014, 2016). D-APPS, in its simplest form, is an analogy "search engine" that returns analogies to the engineer based on the specific performance parameters and critical chain models of the design problem and the analogical solutions. The analogies generated are intended to inspire avenues for design improvements based upon the critical chain models (i.e., crucial chains from the function structure) and desired design performance metrics. To do so, D-APPS uses critical chain models to identify analogies within a repository of critical chain models (the D-APPS design repository). To obtain the critical chain model, the D-APPS approach defines key components of the function structure enabling the transformation of the function structure into a critical chain through the application of expert knowledge by the user. The expert knowledge concepts applied in this transformation process include critical functionality, critical flows, and critical chains as defined by Lucero (2014). These concepts are elaborated on below.

#### 2.1.1. *Critical function*

Not all the functions within the functional model have the same level of significance to the performance of the design. Functions vital to the effective performance of the design are termed *critical functions*. Selecting an appropriate form solution for these functions significantly affects the performance of the overall design. Critical functions are functions that are significantly related to the performance of the design (Lucero et al., 2016) and thus help identify the functions within the critical chain model.

#### 2.1.2. *Critical flows*

Certain material, energy, or signal flows associated with the critical functions are modified by the critical functions re-

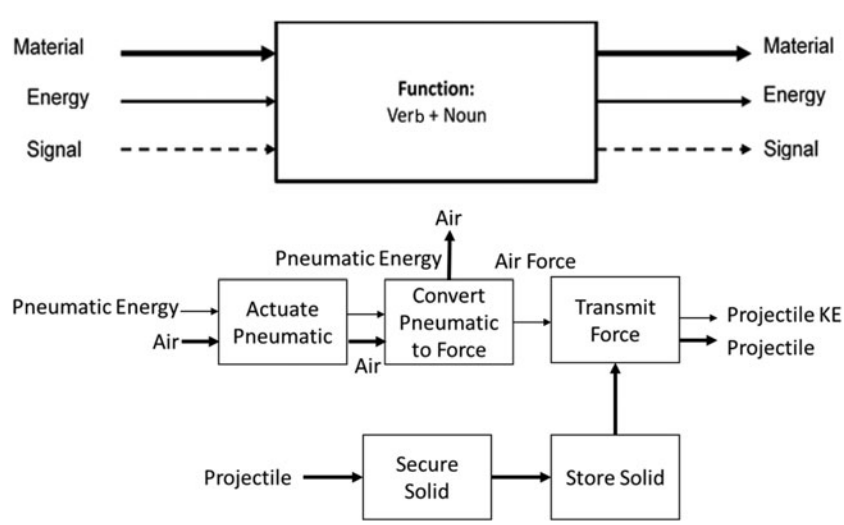


**Fig. 1.** (a) Generic black-box functional modeling. (b) Abstracted function model for a SuperMaxx Ball Shooter.

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

sulting in key performance parameters by which system performance is evaluated. These performance parameters are often (but not always) the target performance metrics for the evaluation of the design as well. The flows that are modified by the critical functions that lead to the key performance parameters are termed *critical flows* (Lucero et al., 2016). The critical flows of a design problem are dependent upon the system functionality and the performance parameters that are the focus of the system design. Just as some functions are more important within the functional model than other functions, some flows are more significant to the performance of the design solution than other flows.

#### *2.1.3. Critical chains*

Combined, critical functions and critical flows define a *critical chain model*. A single function model may be composed of multiple critical chain models, connected by noncritical functions and flows. The critical chain models represent an opportunity to identify design analogies based on elements of function, flow, and performance. Furthermore, these critical chain models can be compared using chain similarity and chain architecture comparative metrics.

### 2.2. Other functional modeling approaches and abstractions

There are other functional abstraction approaches, such as the Idea-Inspire software, which uses a SAPPhIRE (state change, action, parts, phenomenon, input, organs, effect) model to abstract the system (Chakrabarti et al., 2005). Similarly, structure–function–behavior models include information such as different device behavior states, components, substances, and structure, as yet another functional abstraction language (Goel et al., 2013). A slightly different approach is applied in the AskNature.org search engine, which matches functions to associated biological strategies (Biomimicry Institute, 2017). Yet another strategy, the Word Tree Design-by-Analogy Method (Linsey et al., 2012) leverages WordNet (see https://wordnet.princeton.edu/) to make direct linguistic matches at different levels of abstraction and domain specificity to functional descriptions. All of these methods have the capability of operating at different levels of abstraction. The key differences in these approaches are the vocabulary and structure of the abstraction language. Conceptually, a critical chain model could be automatically extracted from each approach with the proper guidelines and rules.

## 3. RESEARCH APPROACH

The D-APPS tool uses critical chain models as the basis to identify potential analogical matches from an analogy database. However, transforming functions structures into critical chains has been a manual, expert knowledge-intensive process, limiting the population of the analogy database. The first goal of this work is to evaluate a number of pruning rules to determine if they represent the basis for an automated approach for transforming functional models to critical chain models. The evaluation is based upon whether the pruning rules lead to a critical chain model that is similar to that obtained by knowledgeable experts. Similarity is measured through a similarity criterion, which compares the automatically generated critical chain models (i.e., pruned functional models) to expert generated critical chain models.

The second part of this research assesses whether critical chain models have value in the identification of design analogies. This assessment is done by generating a set of critical chain models from a collection of functional models. Additional critical chain models representing known (documented) instances of design analogies were added to this set. The critical chain models were then cross-validated to all other members of the set (ignoring self-comparison) and evaluated with the similarity criterion and an additional set of architectural criteria proposed by Morgenthaler (2016) and developed to account for the architectural configuration of the critical chain models. In this assessment, the average critical chain model criterion score is compared for the set of critical chain models that represent known design analogies to answer the question of whether there is a statistically significant difference in criteria value between the analogy and nonanalogy sets.

### 3.1. Pruning rules for functional models

Manually identifying the critical chain model from a single functional model is not an onerous task; however, the identification of critical chain models within a database of analogies totaling hundreds or thousands of examples is a significant undertaking. A means of automating the critical chain model is not only highly desirable but also necessary. Therefore, this paper examines whether pruning rules (Caldwell & Mocko, 2008; Gill et al., 2016) can be used to transform functional models into critical chain models that are consistent to those produced using expert knowledge.

The pruning rules are classified into three different groups: vocabulary, grammar, and topology, as seen in Table 1 (Caldwell & Mocko, 2008). The pruning rules were developed to provide a formalized procedure for functional decomposition in reverse engineering. Caldwell and Mocko (2008) investigated the similarity of the proposed pruning rules to the desired decomposition of the design and noted that the formalized rules provided better insight to achieve desirable decompositions for reverse-engineered designs. These rules were subsequently used to estimate design price predictions (Gill et al., 2016) while the design is at an early design stage. In both studies, these rules were observed to identify the functions with the highest information content within the functional model.

Examination of the residual functions that remained after pruning suggested that there is a similarity to the process of transforming a functional model to a critical chain model. To evaluate this observation, a set of functional models were selected from Otto and Wood (2000), and were subsequently transformed into critical chain models. This transformation was accomplished using expert knowledge (manually) and

**Table 1.** *Classification of composition rules according to Gill et al. (2016) and Caldwell and Mocko (2008)*

| Rule | Composition Rule | Pruning Classification |
|---|---|---|
| CR1 | Remove all import and export functions. | Vocabulary |
| CR2 | Remove all channel, transfer, guide, transport, transmit, translate, rotate, and allow DOF functions referring to any type of energy, signals, or human material. | Grammar |
| CR3 | Remove all couple, join, and link functions referring to any type of solid. | Grammar |
| CR4 | Remove all support, stabilize, secure, and position functions. | Vocabulary |
| CR5 | Remove all control magnitude, actuate, regulate, change, stop, increase, decrease, increment, decrement, shape, condition, prevent, and inhibit functions. | Vocabulary |
| CR6 | Remove all provision, store, supply, contain, and collect functions referring to any type of energy or signal. | Grammar |
| CR7 | Remove all distribute functions referring to any type of energy. | Grammar |
| CR8 | Remove all signal, sense, indicate, process, detect, measure, track, and display functions. | Vocabulary |
| CR9 | Combine adjacent convert functions if the output flows of the first function block are identical to the inputs of the second function block. | Topology |

through the application of the pruning rules (from Table 1). The resulting critical chain models were evaluated for their similarity using a similarity metric established by Morgenthaler (2016).

### 3.2. The similarity metric

To discuss the metric formulation, consider the critical chain model shown in Figure 2. In this example, colored shapes have been substituted in place of the functions from the functional model.

The similarity metric developed by Morgenthaler (2016) measures the similarity between the members of the function sets in two models. Two models with the exact same set of functions, or a self-comparison of models exhibit perfect similarity, as defined in Eq. (1), and represent models with identical membership. However, perfect similarity is not required, as multiple valid models may exist depending upon where the boundary of the model is defined. For instance, the right example in Figure 3 exhibits partial similarity (a shared subchain) while the left example exhibits perfect similarity. Note that the similarity metric does not consider the order of the functions, just the functions that are members of the chain. Hence, the left pair of chains both include the same three functions (red circle, yellow diamond, and blue square). The greater the similarity between two chains, the closer the match between two chains. Note that perfect similarity is not required for an analogy to exist.

$$ChainA \cap ChainB = ChainA = ChainB, \qquad (1)$$

where *ChainA* is the chain of a red circle, yellow diamond, and blue square and *ChainB* is the chain of a yellow diamond, blue square, and a red circle.

Partial similarity can mean that only one function is shared between two chains. Conceptually, even a total lack of similarity can exist, due to conceptual relationships between descriptions in the revised functional basis. Thus, similarity is only one approach for comparing critical chain models. Additional metrics developed by Morgenthaler (2016) take into account the organization of the components of the models, and thus are known as architectural criteria. The metric,

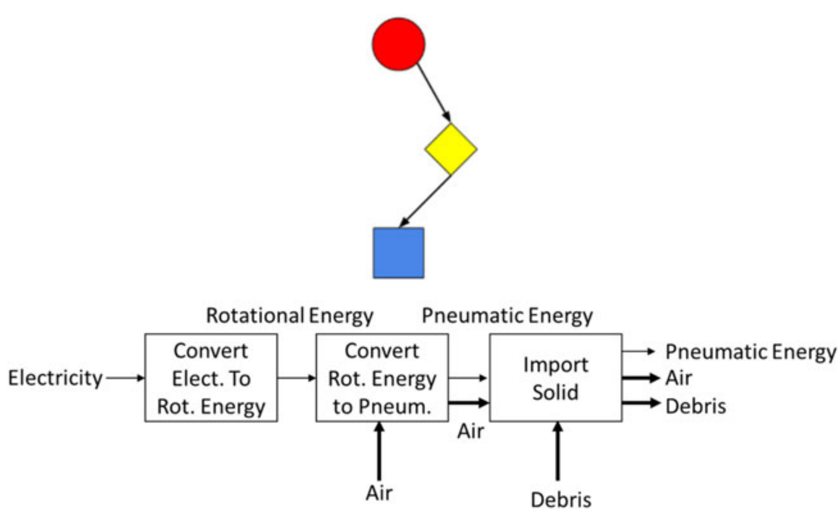


**Fig. 2.** (a) An example of a critical chain where the shape and color of the block relates to a specific function from the revised functional basis. (b) For instance, the red circle would represent the convert electrical energy to rotational energy functional block from the hand vacuum critical chain representation, the yellow diamond would represent the convert rotational energy to pneumatic energy functional block, and the blue square would represent the import solid functional block.

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

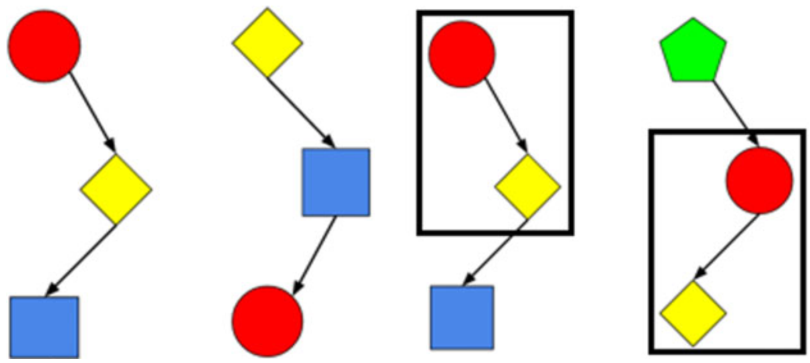

**Fig. 3.** The critical chain models on the left exhibit perfect similarity (all functions exist in both chains) but the critical chain models on the right exhibit only partial similarity (both models share a common subchain).

*similarity*, measures the similarity of two chain models, and because chain models may be of different lengths in the comparison, *similarity* is defined in Eq. (2) as

$$Similarity = \frac{2(\mathrm{FcnShared})}{\mathrm{LC}_1 + \mathrm{LC}_2}, \qquad (2)$$

where FcnShared is the number of functions two chains have in common, $\mathrm{LC}_1$ is the total chain length of the input, and $\mathrm{LC}_2$ is the total chain length of the source required to cover all common functions.

The similarity of the expert knowledge critical chain models to those obtained from applying pruning rules to functional models was compared with three approaches:

1. comparison to the unpruned functional model to functional models pruned by various rule groups, as defined in Table 1,
2. by additionally pruning the function model to only consider the chain of functions connected by the critical flow(s) of the functional model after the application of pruning rules, and
3. by first pruning the functional model to the functions connected by critical flow(s) and then applying pruning rules.

These critical chain models were "highlighted" using an operation defined in Morgenthaler (2016) to disregard excess portions of the model when evaluating the similarity to the input chain model. In this study, the critical chain models transformed via expert knowledge applied to the functional models acted as the input chains, while the functional models that were pruned according to the rules in Table 1 acted as the source chain. The highlighting operation focuses the model comparison on the elements that lie within the span of the chain defined by the input model. This disregards functions that are before or after the span of the input chain, thus evaluating the subset of the model that is comparable to the input reference model. A visual representation of this function chain highlighting operation can be seen in Figure 4.

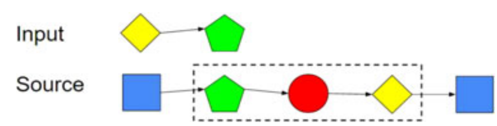


**Fig. 4.** Function chain highlighting example.

The highlighting operation considers the green pentagon and yellow diamond significant because of their commonality to the input chain as well as the red circle or the function that falls in between the common functions. Both of the blue squares would be discarded, as they are not perceived to have significance.

### 3.3. Critical chain extraction experiment

For the critical chain transformation experiment, 23 functional models, listed in Table 2, were selected from Otto and Wood (2000) to form the experimental data set.

The function models were transformed into critical chain models using expert knowledge by two independent experts. Both experts were familiar with the process and terms defined by Lucero (2014). The two experts choose the same basic functional chain in 18 out of 23 models, or an interrater agreement of 78%. For each functional model, the critical functions,

**Table 2.** *List of the 23 products represented with functional models*

| | |
|---|---|
| Ice tea/coffee maker | Fruit/veggie peeler |
| Coffee maker | Engraver |
| Cordless screwdriver | Tailgate |
| Palm sander | Weed trimmer |
| Hand vacuum | Visor |
| Pencil sharpener | Ball shooter |
| Electric knife | Battery |
| Hot air popcorn popper | Battery charger |
| Expresso maker | Drill |
| Sander | Saber saw |
| Wizard | Screw driver |
| Brush | |

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

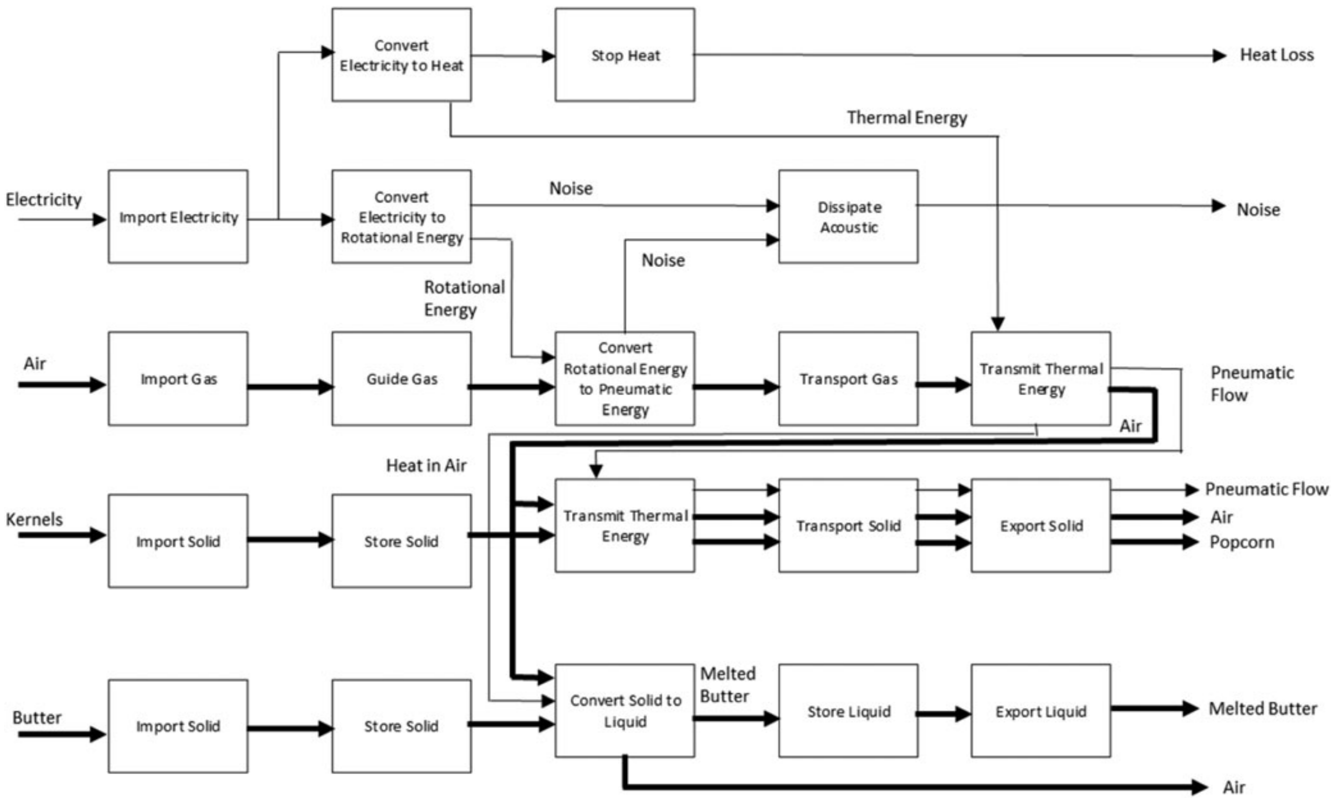

**Fig. 5.** Functional model for hot air popcorn popper adapted from Otto and Wood (2000).

flows, and corresponding critical chains were identified for the primary function of the device. For example, the functional model for a hot air popcorn popper is shown in Figure 5 (Otto & Wood, 2000).

The critical functions identified from the hot air popcorn popper function structure were the following:

1. convert electrical energy to heat
2. convert electrical energy to rotational energy
3. convert rotational energy to pneumatic energy
4. transmit thermal energy
5. transport solid

The critical flows identified from the hot air popcorn popper function structure were the following:

1. electricity
2. air
3. popcorn (both unpopped kernels and popped popcorn)

The set of the critical functions and flows for the hot air popper led to the development of the critical chain model seen in Figure 6. This procedure was repeated for all 23 function structures, and a set of 23 critical chain models, corresponding to the original 23 functional models, were identified.

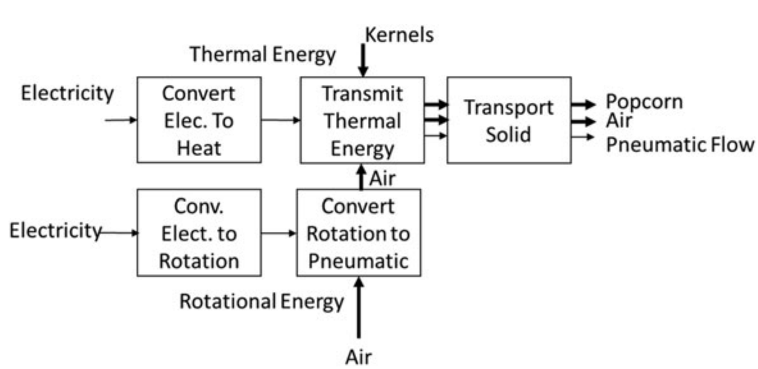

**Fig. 6.** Critical chain developed from hot air popper function structure.

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

**Table 3.** *Similarity metric performance for 18 critical chain models*

| Rules Applied | Via Pruning Rules Only<br>Average ± SD | Via Pruning Rules Followed By Critical Flow Rule<br>Average ± SD | Via Critical Flow Rule Followed By Pruning Rules<br>Average ± SD |
|---|---|---|---|
| None | 0.381 ± 0.080 | 0.848 ± 0.133 | 0.848 ± 0.133 |
| After Rule 1 | 0.452 ± 0.077 | 0.510 ± 0.087 | 0.875 ± 0.117 |
| After Rule 2 | 0.407 ± 0.094 | 0.475 ± 0.094 | 0.858 ± 0.124 |
| After Rule 3 | 0.399 ± 0.093 | 0.475 ± 0.088 | 0.848 ± 0.133 |
| After Rule 4 | 0.406 ± 0.086 | 0.483 ± 0.086 | 0.848 ± 0.133 |
| After Rule 5 | 0.382 ± 0.086 | 0.458 ± 0.087 | 0.833 ± 0.147 |
| After Rule 6 | 0.384 ± 0.081 | 0.462 ± 0.101 | 0.853 ± 0.133 |
| After Rule 7 | 0.379 ± 0.080 | 0.451 ± 0.084 | 0.851 ± 0.137 |
| After Rule 8 | 0.405 ± 0.103 | 0.480 ± 0.092 | 0.848 ± 0.133 |
| After Rule 9 | 0.367 ± 0.090 | 0.449 ± 0.091 | 0.832 ± 0.141 |
| After vocab (Rules 1, 4, 5, & 8) | 0.547 ± 0.105 | 0.609 ± 0.098 | 0.865 ± 0.134 |
| After grammar (Rules 2, 3, 6, & 7) | 0.436 ± 0.117 | 0.505 ± 0.123 | 0.854 ± 0.128 |
| After topology (Rule 9) | 0.367 ± 0.090 | 0.449 ± 0.091 | 0.832 ± 0.141 |
| After vocab & grammar (Rules 1–8) | 0.758 ± 0.148 | 0.776 ± 0.155 | 0.869 ± 0.125 |
| After vocab & topology (Rules 1, 4, 5, 8, & 9) | 0.526 ± 0.112 | 0.602 ± 0.103 | 0.846 ± 0.140 |
| After grammar & topology (Rules 2, 3, 6, 7, & 9) | 0.421 ± 0.131 | 0.499 ± 0.128 | 0.843 ± 0.128 |
| After vocab–grammar–topology (Rules 1–9) | 0.732 ± 0.166 | 0.769 ± 0.158 | 0.887 ± 0.122 |

*Note:* Comparisons are made between the automatically generated critical chain models and the expert knowledge critical chain models. Comparisons are made via pruning rules only, via pruning rules followed by the critical flow rule, and via the critical flow rule followed by the pruning rules.

Subsequently, the 23 original functional models (Otto & Wood, 2000) were pruned using the pruning rules, individually and in six combinations. The pruning rule combinations include

1. unpruned functional models
2. grammar pruned functional models
3. verbal pruned functional models
4. verbal + topology pruned functional models
5. verbal + grammar pruned functional models
6. verbal + grammar + topology pruned functional models

In addition, another pruning rule (the critical flow rule) that eliminated any function that did not carry a critical flow also was considered individually, both before the other rules were applied and after the other rules were applied. With these critical chain models transformed from functional models via the application of pruning rules defined, the similarity metric was calculated between the expert produced critical chain models and the critical chain models obtained via pruning. A statistical analysis was done to indicate the similarity level between these critical chain models.

### 3.4. Results: Identifying an approach for automation

Of the 23 functional models transformed into critical chain models through expert knowledge there was strong interrater agreement for 18 of the critical chain models. The critical chain models for these 18 functional models were then evaluated using the similarity criterion, and the results are shown in Table 3 and a graphical representation is presented in Figure 7.

The results show clear trends. The significance of the critical flow rule for pruning the function structure is clearly demonstrated by the blue hatched bars (third in each group) all achieving higher similarity scores. Furthermore, the yellow solid bars (second in each group) all outperform the red solid bars (first in each group), indicating that applying the critical flow rule provides increased similarity. This also can be observed in the last two column groups of Table 3. Furthermore, when the pruning rules are applied, the similarity is increased when the verbal and grammar rules are applied, but no single verbal, grammatical, or topological rule provides a significant contribution toward achieving similarity, as indicated by the relatively consistent similarity values in each of the sets associated with a single rule. Instead, it is the combination of these rules that lead to similar critical chain models as found through expert knowledge (0.732 for vebal–grammar–topology applied to the functional model compared to having just the critical flow with a similarity value of 0.848). Furthermore, it appears that pruning Rule 9, the only topology rule studied, negatively affects the similarity of the resulting critical chain model unless the critical flow rule is applied first. Rule 9 is used to combine convert functions. Convert functions are often (but not always) critical functions (as found by Lucero, 2014) and thus we hypothesize that the premature combination of convert functions by this rule can result in the elimination of functions that are members of the critical chain model.

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

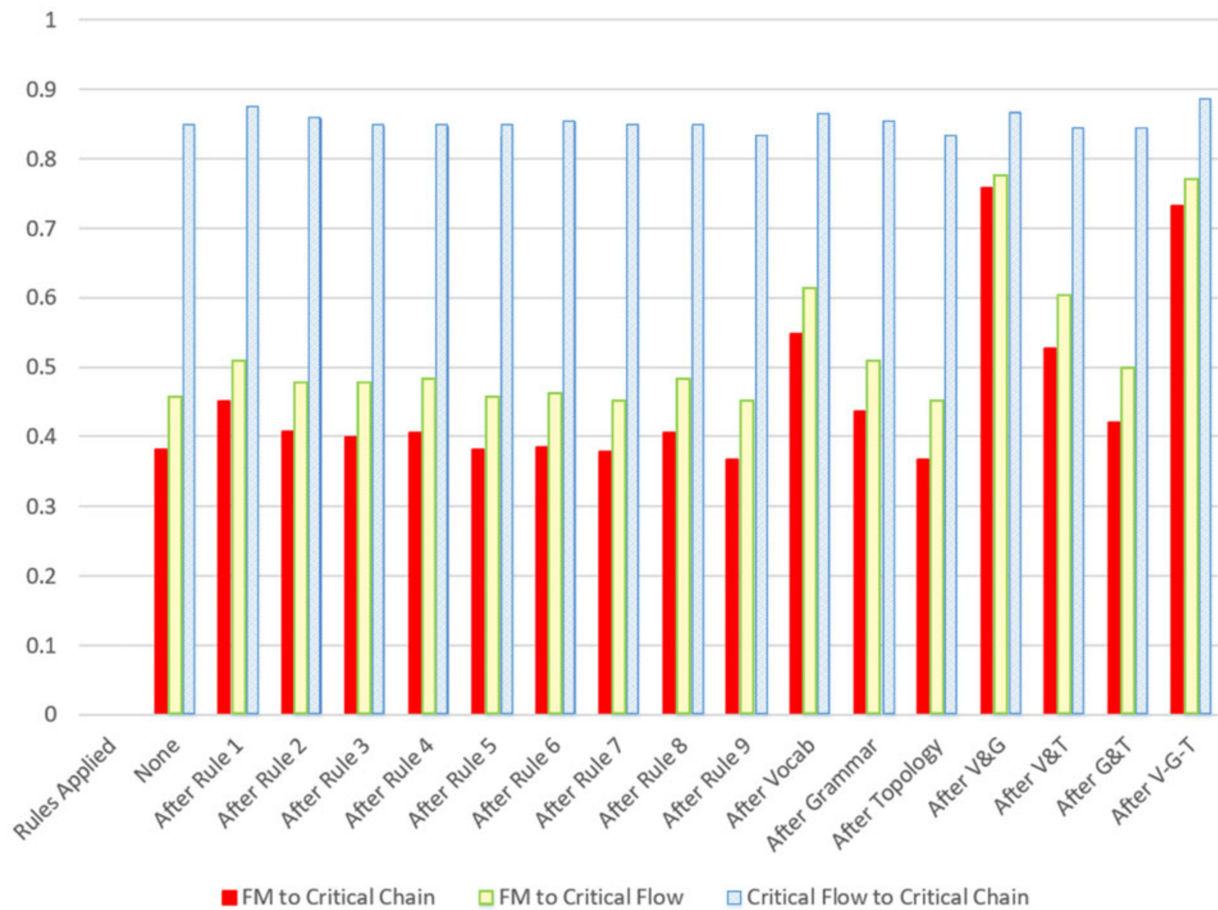


**Fig. 7.** Average similarity metric performance for 18 selected critical chains.

## 4. MATCHING ANALOGIES WITH CRITICAL CHAINS

The identification of critical chain models is used in D-APPS to identify potential analogies from an analogy repository of critical chain models. This section focuses on whether critical chain models can be compared through criteria to yield design analogy matches.

### 4.1. Matching criteria

Critical chain models can be matched not only on the basis of similarity (the common membership of functions within the chain models) but also by their architecture based on a metrics developed by Morgenthaler (2016). Morgenthaler demonstrated that these criteria measure similarities based on their associated architectures, and provided the preliminary data that demonstrated they were effective in analogy identification. This work extends that study with further analysis of the relationships revealed by those criteria.

Architectural criteria provide a measure of the commonality of the order of the functions. The functional chain model examples in Figure 8 exhibits perfect similarity but distinctly different chain model architectures. Even simple linear chain models composed of three or four functions can exhibit a number of distinct architectures including: identical, mirror, disordered, mirrored disordered, and unique. Taken as a whole, the left example in Figure 3 exhibits a disordered architecture where the yellow diamond precedes the blue square, but the red circle does not exhibit a common relationship to the other functions. The subchain of the yellow diamond preceding the blue square is identical in architecture, just as is the right example red circle followed by the yellow diamond in the right example. Figure 8 provides examples of the different architectures.

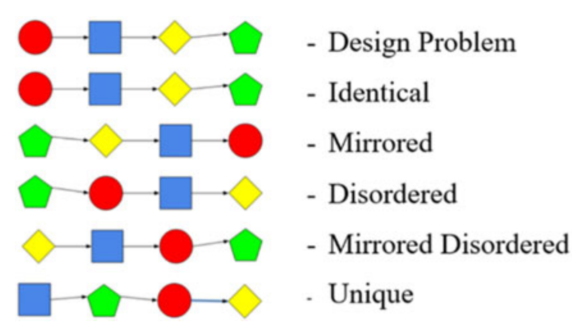


**Fig. 8.** Chain architecture examples.

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

A study of different chain model architectures reveals that even for simple linear chain models, multiple architectures may exist. Furthermore, critical chain models may also exhibit additional nonlinear topologies potentially including trees and rings. Some of these architectural forms result in very close analogies as the order of functions in a functional model is not necessarily unique. This property of functional models is rarely used and is poorly exploited within functional models.

The first architectural metric, *identical*, shown in Eq. (3), is nearly identical to the *similarity* metric, with the exception that its numerator is based on whether or not the chain models share the same function order. If the functions in the same location in the chain models are the same, the FcnSharedOrder is 1; otherwise it is 0. Thus, if the functions do not share an identical order, the metric value is zero. This evaluation begins with the first shared function in the chain models.

$$Identical = \frac{2(\prod \mathrm{FcnSharedOrder})}{\mathrm{LC}_1 + \mathrm{LC}_2}. \tag{3}$$

Similarly, the calculation for the *mirrored* metric, Eq. (4), is also nearly the same as that in Eq. (3). However, in this metric, the FcnSharedInverse term compares the $i$th function to the $m-i$th function in the chain where $m$ is the length of the chain model. If the terms are the same, the expression is equal to 1; otherwise, its value is 0. Thus, the metric is 1 if and only if the chains have the same number of terms in opposite orders.

$$Mirrored = \frac{2(\prod \mathrm{FcnSharedInverse})}{\mathrm{LC}_1 + \mathrm{LC}_2}. \tag{4}$$

The *disordered* and *deredrosid* (i.e., *mirror disordered*), Eqs. (5) and (6), metrics assign a value to the location of each shared function from the input chain model (IFP) to the source chain model (SFP), resulting in the average position differences the two chain models.

$$Disordered = \prod^{n} 1 - \frac{|\mathrm{IFP}_i - \mathrm{SFP}_i + 1|}{n}, \tag{5}$$

$$Deredrosid = \prod_{i}^{\substack{i\\n}} 1 - \frac{|n - \mathrm{IFP}_i - \mathrm{SFP}_i + 1|}{n}, \tag{6}$$

where $n$ is the number of matched functions, $\mathrm{IFP}_i$ is the position of input function $i$, and $\mathrm{SFP}_i$ is the position of source function $i$.

The last metric is the *unique* metric, which is based upon the average of the disordered and deredrosid metrics as shown in Eq. (7).

$$Unique = 1 - \frac{Disordered + Deredrosid}{2}. \tag{7}$$

All of the metrics range from 0 to 1 and represent an initial attempt to measure similarity and architecture between functions in critical chains. Similar efforts also can be developed to incorporate flows in the evaluations. These are certainly not the only architectural metrics that can be derived. However, they represent an initial set of metrics that encompass the observed architectures in critical chain models of this size.

## 4.2. Criteria study

The effectiveness of similarity and architecture comparisons between critical chain models can be evaluated using the criteria defined previously. Through studies of prior analogy implementations such as Ngo (2014) and Ngo et al. (2014), and through the identification of previously identified analogies, a set of 26 critical chain models (Morgenthaler, 2016) representing a total of 59 cases of implemented analogies (some chain models lead to more than one analogy implementation) were identified. Using additional critical chain models from other functional models, an additional 1711 chain model pair comparisons were evaluated (Morgenthaler, 2016). These chain model comparisons predominantly represent nonanalogous design solutions, although no effort was made to filter out unidentified analogies from this set. Using Morgenthaler's (2016) criteria, an exhaustive statistical study of these matches revealed that similarity and architecture metrics do produce positive responses enabling analogy identification.

## 4.3. Study results

Our evaluation of these criteria consisted of an evaluation of known analogies versus simply random chain model

**Table 4.** *Metric performance versus a random set of chain models, a set of known analogy chain models, and the differences between the metric averages*

| | Criteria | | | | |
|---|---|---|---|---|---|
| | Similarity | Identical | Disordered | Deredrosid | Unique |
| Random average ± SD | $0.535 \pm 0.209$ | $0.287 \pm 0.300$ | $0.716 \pm 0.397$ | $0.448 \pm 0.257$ | $0.202 \pm 0.110$ |
| Analogy average ± SD | $0.704 \pm 0.194$ | $0.481 \pm 0.362$ | $0.906 \pm 0.208$ | $0.523 \pm 0.116$ | $0.247 \pm 0.062$ |
| Difference of averages | +0.169 | +0.194 | +0.190 | +0.075 | +0.045 |
| 95% confidence interval | 0.088–0.250 | 0.077–0.311 | 0.091–0.390 | 0.024–0.174 | 0.003–0.087 |
| Two-tailed $p$ | 0.0001 | 0.0011 | 0.0002 | 0.138 | 0.0376 |

*Note:* Adapted from Morgenthaler (2016).

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

comparisons. If the metrics are detecting analogies, then their averages should deviate from the average of chain model comparisons as shown in Table 4.

Examination of Table 4 shows that the similarity, identical, disordered, and unique criteria all show averages that positively deviate from the mean criterion value in a statistically significant manner (two-tail $p < 0.05$). While the deredrosid criterion does not reach this level in this study, there does seem to be some correlation in the deviation from the mean. This higher value may be due to the reduced number of analogy examples that fit into this category within the set of known design analogies.

Unfortunately, in the set of known analogical chain model comparisons, a mirrored analogy example was not included in the study, although we have observed chain models with mirrored architectures during our research. Therefore, we do not have valid results to present concerning the mirrored criterion, and so it was omitted from Table 4. Further research into the analogies within the random sample that appear to be previous unidentified analogy matches is still needed to better understand and to further refine these comparative criteria. However, it is our conclusion, based on the data in Table 4, that the criteria provide a measure of the presence of potential design analogies.

## 5. CONCLUSIONS AND FUTURE WORK

Based on this research, the use of critical chain models as an abstraction tool and the basis for identifying and matching analogies before deabstraction appears to be a promising approach. Table 4 clearly indicates that there is statistical significance for several of the proposed chain model comparison metrics proposed by Morgenthaler (2016). Furthermore, the significant concern that critical chain models can be obtained only through the manual application of expert knowledge appears to be unwarranted. While many of the pruning rules used in prior research did not result in the desired transformations necessary to obtain a similar critical chain model as produced by expert knowledge, the identification of the significance of the critical flow pruning rule as an initial transformation method is a significant outcome of this research. Similar transformations should be possible from alternative functional abstraction models, which will further extend the utility of critical chain models in the identification of design analogies. In addition, the formulation of functional models, and of alternative types of functional models exhibit varying use of grammar, syntax, and levels of abstraction. Understanding and employing these stylistic differences will be important in the continued development of analogy matching tools based upon this abstraction approach.

There were significant improvements in similarity when the critical flow rule was applied as an initial transformation step to convert a functional model to a critical chain model. This supports the contention that the critical flow is just as important as the critical function when looking to build analogies to improve performance. Critical flows are significant because the flow is what is transformed within the functions and therefore form the basis for performance metrics within the design.

Flow is clearly significant in the transformation of functional models into critical chain models. However, individual pruning rules also may be useful. Both the verbal and grammar pruning categories include multiple rules that may add value in performing this transformation. Further research into pruning rules, and perhaps the development of additional pruning rules is likely merited. Furthermore, the definition of critical chain models may not yet be completely defined. It is entirely possible that the chain models revealed by the pruning rules are more complete than that defined through expert knowledge as used in D-APPS. Further studies are necessary.

Once extracted, the combination of similarity and architectural criteria provide a valid and significant method for the comparison of functional chains. Further research into more complex architectures, and into the interactions between criteria, is merited.

## ACKNOWLEDGMENTS

Partial support for this work was provided by National Science Foundation Grants CMMI-1234859 and CMMI-1304383. Comments and opinions herein are those of the authors and do not represent those of the National Science Foundation, the Georgia Institute of Technology, Clemson University, or the Colorado School of Mines.

## REFERENCES

Ball, L.J., Thomas, N.J.M., & Ormerod, C. (2004). Spontaneous analogising in engineering design: a comparative analysis of experts and novices. *Design Studies 25(5)*, 495–508.

Caldwell, B.W., & Mocko, G.M. (2008). Towards rules for functional composition. *Proc. ASME 2008 Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, pp. 319–328. New York: ASME.

Casakin, H., & Goldschmidt, G. (1999). Expertise and the use of visual analogy: implications for design education. *Design Studies 20*, 153–175.

Chakrabarti, A., Sarkar, P., Leelavathamma, B., & Nataraju, B. (2005). A functional representation for aiding biomimetic and artificial inspiration of new ideas. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 19(2)*, 113–132.

Chan, J., & Schunn, C. (2014). The impact of analogies on creative concept generation: lessons from an in vivo study in engineering design. *Cognitive Science*. Advance online publication. doi:10.1111/cogs.12127

Christensen, B.T., & Schunn, C.D. (2007). The relationship of analogical distance to analogical function and preinventive structure: the case of engineering design. *Memory and Cognition 35(1)*, 29–38.

Dieter, G.E., & Schmidt, L.C. (2009). *Engineering Design*, 4th ed. St. Louis, MO: McGraw-Hill.

Gill, A.S., Turner, C.J., & Summers, J.D. (2016). Impact of level of detail and information content on accuracy of function structure-based market price prediction models. *Proc. IDETC DETC* 2016 *Conf.*, Charlotte, NC, August 21–24.

Goel, A.K., McAdams, D.A., & Stone, R.B. (2013). *Biologically Inspired Design: Computational Methods and Tools*. London: Springer Science & Business Media.

Hey, J., Linsey, J., Agogino, A.M., & Wood, K.L. (2008). Analogies and metaphors in creative design. *International Engineering Education 24(2)*, 283–294.

Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., & Wood, K.L. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design 13(2)*, 65–82.

Linsey, J.S., Markman, B., & Wood, K.L. (2012). Design by analogy: a study of the WordTree method for problem re-representation. *Journal of Mechanical Design 134(4)*, 04109.

https://doi.org/10.1017/S0890060417000488 Published online by Cambridge University Press

Lucero, B.M. (2014). *Design-Analogy Performance Parameter System (D-APPS)*. Golden, CO: Colorado School of Mines.
Lucero, B.M., Linsey, J., & Turner, C. (2016). Frameworks for organizing design performance metrics. *Journal of Engineering Design 27(4–6)*, 175–204. doi:10.1080/09544828.2015.1135235
Lucero, B.M., Viswanathan, V.K., Linsey, J.S., & Turner, C.J. (2014). Identifying critical functions for use across engineering design domains. *Journal of Mechanical Design 136*, 121101-1-11.
Mathieson, J.L., Wallace, B.A., & Summers, J.D. (2013). Assembly time modelling through connective complexity metrics. *International Journal of Computer Integrated Manufacturing 26(10)*, 955–967.
Morgenthaler, P. (2016). *Analogy Matching With Function, Flow and Performance*. Golden, CO: Colorado School of Mines.
Nagel, R.L., & Bohm, M. (2011). On teaching functionality and functional model in an engineering curriculum. *Proc. ASME Int. Design Engineering Technical Conf.*, Washington, DC, August 28–31.
Namouz, E.Z., & Summers, J.D. (2013). Complexity connectivity metrics—predicting assembly times with abstract assembly models. In *Smart Product Engineering* (Abramovici, M., & Stark, R., Eds.), pp. 77–78. Berlin: Springer.
Namouz, E., & Summers, J.D. (2014). Comparison of graph generation methods for structural complexity based assembly time estimation. *Journal of Computing and Information Science in Engineering 14(2)*, 021003.
Ngo, P. (2014). *Surveying trends in analogy-inspired product innovation.* MS Thesis. Georgia Institute of Technology.
Ngo, P., Turner, C., & Linsey, J. (2014). Identifying trends in analogy usage for innovation: a cross-sectional product study. *Journal of Mechanical Design 136(11)*, 111109-1-13.
Otto, K., & Wood, K. (2000). *Product Design: Techniques in Reverse Engineering and New Product Development*. Upper Saddle River, NJ: Prentice Hall
Owensby, J.E., & Summers, J.D. (2014). Assembly time estimation: assembly mate based structural complexity metric predictive modeling. *Journal of Computing and Information Science in Engineering 14(1)*, 11004.
Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.H. (2007). *Engineering Design: A Systematic Approach*. London: Springer.
Qian, L., & Gero, J.S. (1996). Function-behavior-structure paths and their role in analogy-based design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10(4)*, 289–312.
Summers, J.D., Miller, M.G., Mathieson, J.L., Mocko, G.M., Summers, J.D., Mathieson, J.L., & Mocko, G.M. (2014). Manufacturing assembly time estimation using structural complexity metric trained artificial neural networks. *Journal of Computing and Information Science in Engineering 14(1)*, 11005.
Vattam, S., Wiltgen, B., Helms, M., Goel, A.K., & Yen, J. (2011). DANE: fostering creativity in and through biologically inspired design. *Proc. Design Creativity 2010*, pp. 115–122. London: Springer.

---

**Malena Agyemang** is a Graduate Research Assistant at Clemson University in the Mechanical Engineering Department. Her research focus is generation and application of design analogies in design methodology and theory. The goal of Malena's work is to investigate the potential of analogous representation in various aspects of design theory and methodology, focusing on engineering design for global development.

**Julie S. Linsey** is an Associate Professor in the George W. Woodruff School of Mechanical Engineering at the Georgia Institute of Technology. Her research focus is on design methods, theory, and engineering education with a particular focus on innovation and conceptual design. The goal of Dr. Linsey's research is to discover new knowledge about how engineers think and leverage this knowledge into design methods and tools to improve engineering design. She has authored over 100 technical publications including over 30 journal papers and six book chapters, and she holds two patents.

**Cameron J. Turner** is an Associate Professor of mechanical engineering and the Founder of the Design Innovation and Computational Engineering (DICE) Laboratory, a part of the Clemson Engineering Design Applications and Research (CEDAR) Group at Clemson University. Cameron earned his doctorate (engineering design) and masters (robotics and automation) at the University of Texas at Austin, and his BSME (thermal-fluids and solid mechanics) at the University of Wyoming. Dr. Turner was previously an Associate Professor of mechanical engineering at the Colorado School of Mines and a Research and Development Engineer and Subject Matter Expert at Los Alamos National Laboratory. His research has been supported by grants from government, large industry, and smaller manufacturing companies. His areas of interest include computational design methods whereby the computer and engineer collaborate during the design and manufacturing process. His work has resulted in computational tools for design analogy identification, tools for design analysis and optimization, and methods for intelligent integration of robotics and automation systems. He is a registered Professional Engineer in the State of New Mexico.