ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

FUNCTION IN ENGINEERING: BENCHMARKING REPRESENTATIONS AND MODELS

Joshua David SUMMERS (1), Claudia ECKERT (2), Ashok GOEL (3)
1: Clemson University, United States of America; 2: Open University, United Kingdom; 3: Georgia Tech., United States of America

ABSTRACT

This paper presents the requirements and needs to establish a benchmarking protocol for systematic comparison of different function modeling representations. This benchmarking protocol includes representation characteristics, supported cognitive dimensions, and enabled reasoning activities. Problem types are also defined as: reverse engineering, familiar products, novel products, and single-component systems. It is recommended that researchers and developers of function modeling representations work together to define a canonically acceptable set of benchmark tests and evaluations so that clear benefits and weaknesses for the disparate collection of approaches can be compared. This paper is written as a call to action for the research community to begin to look at establishing a benchmarking standard protocol for function modeling comparison purposes. This protocol should be refined with input from developers of the competing approaches in an academically open environment.

Keywords: function, function modeling, reasoning, representation

Contact:
Prof. Joshua David Summers
Clemson University
Mechanical Engineering
Clemson
29634-0921
United States of America
JSUMMER@CLEMSON.EDU

1 FUNCTION REPRESENTATIONS IN ENGINEERING

Reasoning about functional aspects of products is critical in product development. Therefore, educators argue for the use of many different functional models to support systematic products development, yet without clear guidance on which approach is appropriate for which reasoning need (Pahl et al., 2007; Ullman, 2010). Therefore, this paper provides a justification and a proposed research direction for establishing a common benchmarking scheme for function representations that are developed and deployed throughout academia and practice with the ultimate goal of providing industry with practically usable functional modeling tools and concepts. Despite decades of research into functional descriptions, industry has not appeared to have incorporated functional modeling in practice while still proclaiming a need to express product information beyond form. Possible reasons contributing to this resistance might be that there is not yet a canonical definition of function, each approach being grounded in different conceptualizations or that there might be multiple distinct concepts with shared terminology. Researchers and practitioners have proposed many different views of function in engineering design (Crilly, 2010; Deng, 2002; Eckert, 2013; V. Srinivasan et al., 2012; Vermaas, 2013). These views have resulted in many different approaches to model information about a product's functionality. For example, several design textbooks talk about using function-flow networks to capture the sequence and dependencies for the desired functionality of a product (Pahl et al., 2007; Ullman, 2010). Rather than develop a single, unified definition of function, we assert that each approach has its own strengths and weaknesses; each approach is useful and are particularly well suited for different reasoning applications and domains yet the transference across these being difficult at best. Therefore, we are proposing a different approach to function research; by developing a set of comparative benchmarks that can be explored with the different modeling approaches, the community can start to discern which approaches are more useful for different needs, and perhaps to discover which elements of the representations and vocabularies are most conducive for different elements of functional thinking.

The information captured within function models can be used to facilitate many different engineering activities across entire product lifecycle, such as synthesis, analysis, exploration, visualization, explanation, and fault detection (J. Gero and Kannengiesser, 2002; Goel and Bhatta, 2004; Kurtoglu and Tumer, 2008). Despite the proliferation of function modeling research in the literature, there does not appear to be a significant adoption in industry (Eckert, 2013). Two assumptions are that the representations are not perceived as easy to use or learn and that the potential users are not well informed with respect to what these representations can provide; both relating back to educational challenges for the community. These two keys need to be addressed with each representation and modeling approach proposed, but with common frames of reference, perhaps supported through a standard benchmarking protocol that would define common problems and common issues against which methods can be challenged.

2 THE NEED FOR BENCHMARKING

Next, one can turn to other areas of research that have developed canonical benchmark systems that have proven to be useful in cross-comparison of algorithms and methods. The Traveling Salesman Problem has been developed into a series of benchmark problems that can be used to compare algorithms (Peterson, 1990). Likewise, the field of optimization have several accepted benchmark problems, again used to evaluate performance of new algorithms (Brest et al., 2006) and available online (Mittelmann, 2012). Alternatively, one also finds standard benchmark tests in the automotive industry when considering different control strategies (Rajamani, 2012).

Each of these different benchmark sets have been constructed to test new algorithms, either optimization or controls. With developing a benchmark set of problems for comparing function representations, the algorithm, or reasoning dimension must be considered. Furthermore, the representation and the modeling of the functions should also be considered; defining a critical distinction between the traditional approaches of benchmarking and the approach proposed here.

There has never been a benchmarking activity in functional modeling and while there are many discussion of how concepts are related there is no systematic comparison of the expressive power of different models. An experiment on functional descriptions, where different engineers were provided with a product and asked to generate a functional description (Eckert et al., 2012), has shown that there was a huge variability between individuals and in particular between different modeling approaches.

There is a need to look systematically at how different modeling approaches compare in representational expressive power, reasoning inferencing capacity, and modeling ease of use.

3 IS FUNCTION RESEARCH SUFFICIENTLY MATURE?

Before defining a series of benchmark test cases to be used in cross-evaluation of competing function representations, one must first ask the question about whether the research field is sufficiently mature to warrant such an effort. This is critical to determining whether there is both sufficient need in a plethora of competing approaches and a sufficient population size of researchers ready and willing to use these benchmarks as comparative tools. To this end, one can first consider the field's evolution over the past five decades (Table 1). This is not intended as a comprehensive survey of function research, but as a quick assessment of the maturity of the field.

Table 1: Recent Decades of Engineering Function Research

Decade	Example References
1960's	(Eastman, 1969; Pahl et al., 2007)
1970's	(Collins et al., 1976; Freeman and Newell, 1971; Rodenacker, 1971)
1980's	(Andreasen and Hein, 1987; Hubka and Eder, 1988; V. Sembugamoorthy et al., 1986; Ullman et al., 1988)
1990's	(Bracewell and Sharpe, 1996; Goel, 1997; Kirschman and G. M. Fadel, 1998; Qian and J.S. Gero, 1996; Sasajima et al., 1995; Umeda et al., 1996; Vescovi et al., 1993)
2000's	(Albert Albers et al., 2008; Chandrasekaran, 2005; Erden et al., 2008; J. Gero and Kannengiesser, 2002; Hirtz et al., 2002)
2010's	(Linz, 2011; Schultz et al., 2010; C Sen et al., 2011; V. Srinivasan et al., 2012; Yang et al., 2010)

While there are many different representations found in the literature, such as the Structure-Behavior-Function (Bhatta and Goel, 1997), the Function-Behavior-Structure (Qian and J.S. Gero, 1996), the Functional Basis and derivatives (Hirtz et al., 2002), the Function-Behavior-State (Umeda et al., 1996), affordances (Jonathan R A Maier et al., 2007), the Connect and Channel Model (Albert Albers et al., 2008), and general function lists used in design tools such as morphological charts (Smith et al., 2012), the House of Quality (Olewnik and Lewis, 2005), and Axiomatic Design (Suh, 1999).

These different function representations and modeling approaches have been developed by many researchers across the world, each with a different intent, history, and context behind the representation. It is because of the many different roles and uses within engineering design that these models have evolved disparately. Unfortunately, many of these approaches have suffered from the "inventors" problem within design research – researchers will push a solution to a problem and a design need without actually designing the tool or method based on the intrinsic properties. Thus, many of the representations, while serving different specific purposes such as machinery and manufacturing systems with an emphasis on flows (Pahl et al., 2007), might support other activities addressed by competing representations. This suggests that there is a need for developing a systematic comparison system.

4 DIMENSIONS OF COMPARISON

We propose that there are three interlinking dimensions of a function modeling approach that should be compared: representation characteristics, supported cognitive dimensions, and enabled reasoning activities. The intent of this paper is not to provide a comprehensive ontology of these dimensions. Rather, we seek to illustrate how these dimensions might be important in the comparison and how different benchmark problems can be used to explore these dimensions explicitly and implicitly.

Representation Characteristics

When comparing function modeling approaches, the typical approach is to start first with comparing the representations at the vocabulary and grammatical levels. Comparisons can be made similar to AI representation comparisons: representational adequacy, inferential adequacy, inferential efficiency, and acquisitional efficiency (Winston, 2005). Another approach to compare representations examines the vocabulary, structure, expression, purpose, and abstraction (J D Summers and Shah, 2004). We propose that the representation comparison could include, but not be limited to:

- *scope*: domain for which the function modeling approach is intended (Nagel et al., 2008)
- *flexibility*: ability to modify and adapt the representation to address new problems (Regli et al., 2000)
- *indexing*: support access to the right (or useful) knowledge when needed (Goel and Bhatta, 2004)

- *consistency*: enforce physics and other consistency (Chiradeep Sen et al., 2011)
- *translationabilty*: tied to other engineering models (Nebel, 2000)
- *behavior:* ability of the representation to simulation behavior (Qian and J.S. Gero, 1996)
- *scalability:* support both simple and complex problem types (Chiang et al., 2001)

Modeling Characteristics

In addition to the representational issues, how the designer interacts with building the model is of concern when comparing the functional modeling approaches. For instance, is the modeling computationally supported, restricted to human effort, or is a mixed initiative approach supported (Chiradeep Sen et al., 2012). Additionally, whether the function modeling supports different types of construction approaches, such as forward chaining (moving from input to output), backward chaining (moving from output to input), nucleation, environment to system (outside to inside), or system to environment (inside to outside) is an important consideration (Chiradeep Sen and Joshua D Summers, 2012). A final characteristic could be related to whether the modeling approach support decomposition and recomposition across multiple hierarchical levels and abstractions (Pahl et al., 2007).

Cognitive Dimension Characteristics

The concept of cognitive dimensions has been developed in Human Computer Interaction to help software designers to think through the usability of the artifacts they were creating, such as programming languages or user interfaces (Green and Petre, 1996), because many developers or software engineers have experience to develop well-design information artifacts, they no way of articulating why they were appropriate to meet user needs. Functional modeling approaches can be seen as information artifacts, similar to programming languages. Cognitive dimensions offer a vocabulary for discussing usability issues, that is informed by cognitive science (Blackwell et al., 2001). The framework is deliberately broad to avoid being overwhelmed in the details of an implementation and thereby losing the sought conceptual improvements. However the approach is also task-specific; addressing processes and activities rather than merely assessing the final product. Therefore it can be used to evaluate *functional modeling approaches* rather than just offer a way to compare functional models. The cognitive dimensions are orthogonal to support reasoning trade-offs and to analyze the space of possible solutions in a coherent manner, and where possible look at the effect of combinations of dimensions. Table 2 presents a selection of the cognitive dimensions with their questions for programming and a possible interpretation of these questions for functional modeling, which would need to be refined prior to bench marking exercise (Green and Petre, 1996).

Table 2: Key cognitive dimensions based on (Green and Petre, 1996)

Dimension	Question for Programming Languages	Question of Functional Modelling
Abstraction Gradient	What are the minimum and maximum levels of abstraction? Can fragments be encapsulated?	What are the minimum and maximum levels of abstraction? Can partial be created?
Closeness of mapping	What 'programming games' need to be learned?	What modeling conventions needs to be learned? How intuitive is the resulting model?
Error-proneness	Does the design of the notation induce 'careless mistakes'?	Does the design of the notation induce 'careless mistakes'?
Hidden dependencies	Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic?	Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic?
Premature commitment	Do programmers have to make decisions before they have the information they need?	Do the model require decisions before they have the information needed is available?
Secondary notation	Can programmers use layout, colour, or other cues to convey extra meaning, above and beyond the 'official' semantics of the language?	Can the models be annotated or linked to other product representations?
Viscosity	How much effort is required to perform a single change?	How much effort is required to perform a single change? How easy is it to adapt the model from a model of a similar product.
Visibility	Is every part of the code simultaneously visible (assuming a large enough display), or is it at least possible to juxtapose any two parts side-by-side at will? If the code is dispersed, is it at least possible to know in what order to read it?	How easy is it to see all aspects of the model? Can two models be compared?

The computing cognitive dimensions have a dimension of the **progressive evaluation** referring to the ability to obtain feedback on the modeling through the process, which appears far more meaningful for a programming language, where a program can be run than a model that is deployed in many different ways. There are also dimensions specifically aimed at the notion like **diffuseness**, which addresses the number of symbols or graphic entities are required to express a meaning, and **hard mental operations**, which questions the need for annotations. These depend very much on a particular implementation version, as few standards around functional modeling have yet emerged.

For each of the dimension a scale of sub categories can be developed. For example abstraction gradient is decomposed into abstraction-hating, abstraction-tolerant, and abstraction-hungry (Green and Petre, 1996). For example, while people might find abstraction-hungry programming languages hard, abstraction can reduce error-proneness and increase viscosity.

Reasoning Characteristics

Reasoning is the comparison dimension that motivates the need for a common, standard benchmark for evaluating function modeling approaches. It is for different classes of reasoning that each function model is constructed. These activities range from failure detection (Kurtoglu and Tumer, 2008), reverse engineering and product understanding (Hirtz et al., 2002), design decision justification (J. Gero, 1996), or concept definition and exploration (Pahl et al., 2007). Some types of reasoning that can be evaluated with respect to support include:

- *interpretability*: how consistent and precise is the interpretation of the function models across different individuals, domain, and expertise (B. Caldwell et al., 2012)
- *physics maintenance:* can questions about conservation of energy or material, irreversibility, or other physics queries be answered (Chiradeep Sen et al., 2011)
- *analogical mapping*: does the representation support analogical mapping and alignment (Qian and J.S. Gero, 1996)
- *pattern learning*: does the representation support learning of abstractions needed for analogical transfer (Bhatta and Goel, 1997)
- *state transformations*: does the representation support answering questions about different states (Deng, 2002)
- *change propagation*: does the representation support discovery about the effects of perturbations in the system (Kurtoglu and Tumer, 2008)

These reasoning dimensions might relate to the cognitive dimensions. For example, interpretability, analogical mapping, and change propagation might relate to closeness of mapping and viscosity. On the other hand, physics maintenance and state transformation is focused more on the content of the model. Other challenges in reasoning might relate to the ability to contextualize the system within a larger environment or the distribution of system level functions to several distributed elements.

5 BENCHMARK PROBLEM TYPES

In order to explore the different characteristics of representation that enable cognitive dimensions to support reasoning activities, a set of benchmark problems are needed. Several problem type classifications are offered in the literature, but we propose here four types for study, an example for each type found in the literature, and a list of alternative examples for each. We notably do not include large scale, complex systems such as submarines, aircraft, or space systems as these are not readily available to all researchers for benchmarking activities. Further, other product type classifications might include application domains, such as automotive vs. consumer electronics vs. power tools, but this is out of scope for this paper and can be visited in future revisions of the benchmark protocols.

Reverse Engineered Products

Many function modeling approaches have been demonstrated on existing products after dissection and reverse engineering. A repository of commercial products, that have been reversed engineered to understand them has been developed with the function representation serving as the foundation for the information model (Bohm et al., 2005). An advantage of including this type of problem in the benchmarking formalism is that the products exist and their performance can be measured and evaluated. A reverse engineered product provides a common platform for comparison. An example product that has been used extensively (Hamraz et al., 2012; Huang and Jin, 2009; Jonathan R. A. Maier et al., 2007) to explore function modeling is the hairdryer (Figure 2 and Figure 1). Other possible products that could be considered might be pneumatic impact drivers (increased amount of mechanical components), battery power tools (readily available in multiple variations), vacuum cleaners (potential to compare across multiple customer cultural differences), or bike lights (simple and inexpensive systems). It is important that a common product be selected so that the community can standardize their demonstration cases.

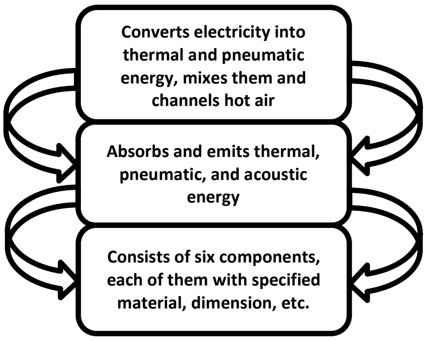

Figure 1: Example Knowledge Types in the Hairdryer (Hamraz et al., 2012)

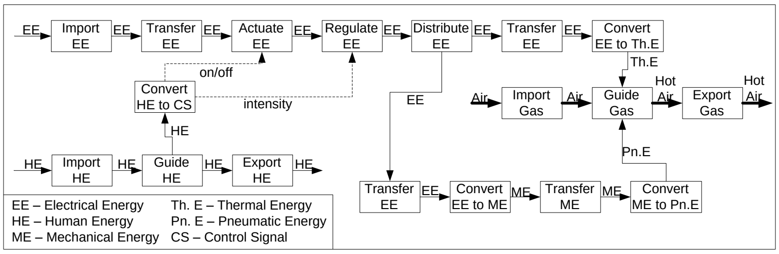

Figure 2: Function structure of a hairdryer product stored in the Design Repository (http://repository.designengineeringlab.org/)

Familiar Product

Reverse engineering and dissecting products allow engineers to map existing systems and components to specific functionality. Often, though, as a first step in reverse engineering is to hypothesize the internal functioning of a product (Otto and K.L. Wood, 1998). Further, while reverse engineering can test the ability of a representation to model the detailed functionality of an existing system, modeling a familiar without having the product in hand can expose the ability to be fluidly and flexibly model the system, as significant backtracking and hierarchical jumping is likely. An example of the results from an experimental exercise to explore how engineers model known products is found in Figure 3 (Eckert et al., 2011). Different engineers are likely to model the system in different ways even given the same underlying representation, so the expressive power of modeling approaches can be assessed. Thus, this benchmark product can be used to explore aspects of the representation, such as consistency and repeatability, without the cost of buying products to reverse engineer. Other products than can be considered might be bicycles, gear boxes, or printing machines.

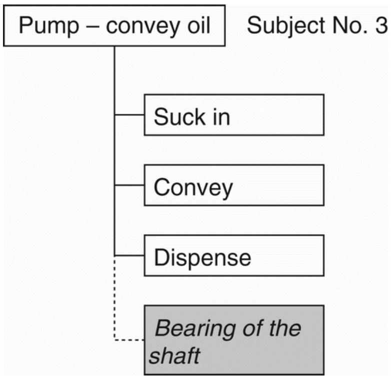

Figure 3: Example Function Model for a Hydraulic Pump (Eckert et al., 2011)

Novel Products

Generative forward system design is where new multi-component systems are developed for problems previously not addressed. In this way, the characteristics include: novelty (not done before), system (more than one component), and intentional (design with a purpose). Examples of “problems” that could be used to benchmark and compare different function modeling approaches include: automated omelet makers, hand cranked pretzel makers, shoe string tying mechanisms, machine to fold clothes for a hotel room, and a hand cranked automated burrito maker. Ideally, benchmark examples could be drawn from literature to support the objectivity of benchmark. For example, the burrito folding system problem has been used in past design experimentation, such as comparing function lists and function structures in morphological charts (Richardson III et al., 2011). Figure 4 illustrates the function lists and structures for the burrito folder that were used ideation experiments. This type of benchmark can help explore the degree to which a function representation can be used in understanding novel problems and generating new solutions.

(store, {filling}, {filling})
(position, {tortilla}, {tortilla})
(fill, {filling, tortilla}, {unfolded})
(fold, {unfolded}, {burrito})
(dispense, {burrito}, {burrito})

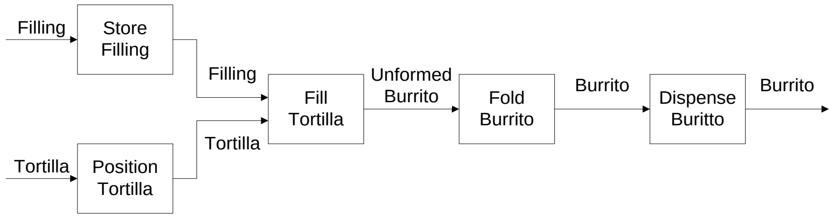

Figure 4: Function List and Function Structure for Burrito Folder (Richardson III et al., 2011)

Single-Component Products

Another example product that can be used as an interesting benchmark case is that of single-component multi-functional products, such as passive morphing airfoils (Schultz et al., 2010) and speed screws (Albert Albers et al., 2008). For example, the design and analysis of a speed screw demonstrated how the Contact and Channel Model can be used in design of a single component. Not all function models might be able to capture the functionality and behaviors associated with single components. This scaling ability, both to large systems and to small systems, should be explored. While the speed screw benchmark example (Figure 5) shows the downward scalability of reverse engineering, the passive, morphing airfoil design (Figure 6) illustrates the downward scalability of forward engineered products. Larger scale systems, such as aircraft are not considered within the benchmarking protocol because of the challenge of general access for the researchers.

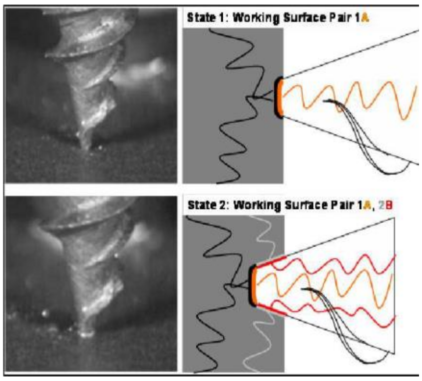

Figure 5: States of WS Pairs for the C&C Model for a Speed Screw (Albert Albers et al., 2008)

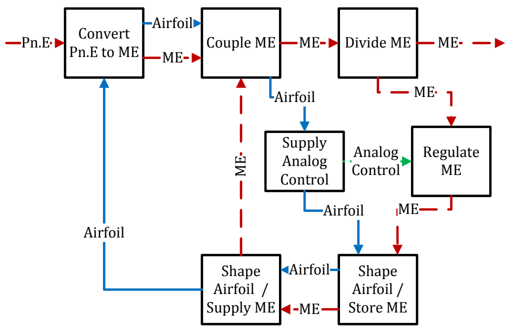

Figure 6: Function Structure for a Morphing Airfoil (Schultz et al., 2010)

6 RESEARCH RECOMMENDATIONS

This paper is written as a call to action for the research community to begin to look at establishing a benchmarking standard protocol for function modeling comparison purposes. This protocol should be refined with input from developers of the competing approaches in an academically open environment. Researchers should find clear value in this benchmarking process as it (1) forces the disparate communities to begin to talk with each other, (2) distributes tutorials on how to develop and execute the variety of models and methods thereby enhancing the education of future engineers, and (3) could be paired with a reasoning/representation selection database to help more systematically develop informed tools and methods. The benchmarking exercises can also help researchers justify a systematic evolution of their approaches.

One approach to achieving this could be to first create a benchmarking development group. This group should be diverse internationally, representing the various different families of approaches. In addition to the intellectual motivation for benchmarking, the group could provide pressure on funding agencies to create benchmarking activities that would include monetary support.

The benchmarking dimensions and characteristics described above are not intended to serve as the final set, but rather to serve as starting point for development. Additional study is needed to refine these characteristics and to strategically select case examples for comparison. The development group can support this directly.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of G-SCOP, CNRS, and INP-Grenoble in supporting Dr. Summers as a visiting professor for 2012-2013.

REFERENCES

Albers, A., Thau, S. and Alink, T. (2008) Support of Design Engineering Activity Through C & CM - Temporal Decomposition of Design Problems, in Horvath, I. (Ed.),*Tools and Methods for Competitive Engineering Conference*, Izmir, Turkey, pp. 295–306.

Andreasen, M. and Hein, L. (1987) *Integrated product development*, IFS (Publications), London, UK.

Bhatta, S.R. and Goel, A.K. (1997) A Functional Theory of Design Patterns, *15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, Vol. 1, p. vol. 1.

Blackwell, A., Britton, C., Cox, A., Green, T., Gurr, C., Kadoda, G., Kutar, M., et al. (2001) Cognitive dimensions of notations: Design tools for cognitive technology, *Cognitive Technology: Instruments of Mind*, Springer, pp. 325–341.

Bohm, M., Stone, R. and Szykman, S. (2005) Enhancing Virtual Product Representations for Advanced Design Repository Systems, *ASME Transactions Journal of Computing and Information Science in Engineering*, Vol. 5 No. 4, pp. 360–72.

Bracewell, R.H. and Sharpe, J.E.E. (1996) Functional Descriptions Used in Computer Support for Qualiative Scheme Generation - "Schemebuilder", *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 10, pp. 333–345.

Brest, J., Greiner, S., Boskovic, B., Mernik, M. and Zumer, V. (2006) Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems, *IEEE Transactions on Evolutionary Computation*, Vol. 10 No. 6, pp. 646–657.

Caldwell, B., Thomas, J., Sen, C, Mocko, G M and Summers, J D. (2012) The Effects of Language and Pruning on Function Structure Interpretability, *Journal of Mechanical Design*, ASME, Vol. 134 No. 6, p. 061001.

Chandrasekaran, B. (2005) Representing Function: Relating Functional Representation and Functional Modeling Research Streams, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 19, pp. 65–74.

Chiang, W.-C., Pennathur, A. and Mital, A. (2001) Designing and manufacturing consumer products for functionality: a literature review of current function definitions and design support tools, *Integrated Manufacturing Systems*, MCB UP Ltd, Vol. 12 No. 6, pp. 430–448.

Collins, J.A., Hagan, B.T. and Bratt, H.M. (1976) The Failure-Experience Matrix—A Useful Design Tool, *Journal of Engineering for Industry*, Vol. 98 No. 3, p. 1074. doi:10.1115/1.3439009

Crilly, N. (2010) The roles that artefacts play: technical, social and aesthetic functions, *Design Studies*, Elsevier, Vol. 31 No. 4, pp. 311–344.

Deng, Y.M. (2002) Function and behavior representation in conceptual mechanical design, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 16 No. 05, pp. 343–362.

Eastman, C.M. (1969) Cognitive Processes and Ill-Defined Problems: A Case Study from Design, *Joint International Conference on Artificial Intelligence*, Washington, DC, pp. 669–690.

Eckert, C. (2013) That which is not form : the practical challenges in using functional concepts in design, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 27 No. 3, p. to appear.

Eckert, C., Alink, Thomas, Ruckpaul, Anne and Albers, Albert. (2011) Different notions of function: results from an experiment on the analysis of an existing product, *Journal of Engineering Design*, Taylor & Francis, Vol. 22 No. 11-12, pp. 811–837.

Eckert, C., Ruckpaul, A, Alink, T and Albers, A. (2012) Variations in functional decomposition for an existing product: Experimental results, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Cambridge Univ Press, Vol. 26 No. 02, pp. 107–128.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E. and Tomiyama, T. (2008) A review of function modeling: Approaches and applications, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 22 No. 02, pp. 147–169. doi:10.1017/S0890060408000103

Freeman, P. and Newell, A. (1971) A Model for Functional Reasoning in Design, *Second International Joint Conference on Artificial Intelligence*.

Gero, J. (1996) Creativity, Emergence, and Evolution in Design, *Knowledge-Based Systems*, Vol. 9, pp. 435–448.

Gero, J. and Kannengiesser, U. (2002) The Situated-Function-Behaviour-Structure Framework, in Gero, J. (Ed.),*Artificial Intelligence in Design*, Kluwer Academic Publishers, Norwell, MA, pp. 89–104.
Goel, A.K. (1997) Design, Analogy, and Creativity, *IEEE Intelligent Systems*, Vol. 12 No. 3, pp. 62–70.
Goel, A.K. and Bhatta, S.R. (2004) Use of Design Patterns in Analogy-Based Design, *Advanced Engineering Informatics*, Vol. 18, pp. 85–94.
Green, T. and Petre, M. (1996) Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework, *Journal of visual languages and computing*, Vol. 7 No. 2, pp. 131–174.
Hamraz, B., Caldwell, N.H.M. and Clarkson, P.J. (2012) A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design, *Journal of Mechanical Design*, Vol. 134, p. 100905.
Hirtz, J., Stone, R., McAdams, D., Szykman, S. and Wood, K. (2002) A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts, *Research in Engineering Design*, Vol. 13, pp. 65–82.
Huang, Z. and Jin, Y. (2009) Extension of stress and strength interference theory for conceptual design-for-reliability, *Journal of Mechanical Design*, Vol. 131 No. 7, p. 71001.
Hubka, V. and Eder, W.E. (1988) *Theory of technical systems: a total concept theory for engineering design, Berlin and New York, Springer-Verlag, 1988, 291 p.*, Springer Verlag, New York, NY, Vol. 1.
Kirschman, C.F. and Fadel, G. M. (1998) Classifying Functions for Mechanical Design, *Journal of Mechanical Design*, Vol. 120 No. 3, p. 475.
Kurtoglu, T. and Tumer, I. (2008) A Graph-Based Fault Identification and Propagation Framework for Functional Design of Complex Systems, *ASME Transactions Journal of Mechanical Design*, Vol. 130 No. 5, pp. 1–8.
Linz, P. (2011) *An Introduction to Formal Languages and Automata*, Jones & Bartlett Publishers, Sudberry, MA, 5thed., p. 437.
Maier, J, Srinivasan, A., Vikram, B., Summers, J. and Bettig, B. (2007) A Computational Framework for Semantically Rich Design Problems Based on the Theory of Affordances and Examplar Technology, *International Conference on Engineering Design*, The Design Society, Paris, France, pp. 205–206.
Maier, J., Ezhilan, T., Fadel, G M, Summers, J D and Mocko, G. (2007) A hierarchical requirements modeling scheme to support engineering innovation, *International Conference for Engineering Design, Paris, France*.
Mittelmann, H. (2012) Decision Tree for Optimization Software, *website*. Retrieved September 13, 2012, from http://plato.asu.edu/bench.html
Nagel, R.L., Vucovich, J.P., Stone, R.B. and McAdams, D.A. (2008) A Signal Grammar to Guide Functional Modeling of Electromechanical Products, *Journal of Mechanical Design*, Vol. 130, p. 51101.
Nebel, B. (2000) On the Compilability and Expressive Power of Propositional Planning Formalisms, *Journal of Artificial Intelligence Research*, Vol. 12 No. 1, pp. 271–315.
Olewnik, A.T. and Lewis, K. (2005) On Validating Engineering Design Decision Support Tools, *Concurrent Engineering*, Vol. 13 No. 2, pp. 111–122.
Otto, K.N. and Wood, K.L. (1998) Product evolution: a reverse engineering and redesign methodology, *Research in Engineering Design*, Springer, Vol. 10 No. 4, pp. 226–243.
Pahl, G., Beitz, W., Wallace, K. and Blessing, L. (2007) *Engineering Design: A Systematic Approach*, Springer-Verlag London Limited, London, 3rded.
Peterson, C. (1990) Parallel Distributed Approaches to Combinatorial Optimization: Benchmark Studies on Traveling Salesman Problem, *Neural Computation*, MIT Press 238 Main St., Suite 500, Cambridge, MA 02142-1046 USA journals-info@mit.edu, Vol. 2 No. 3, pp. 261–269.
Qian, L. and Gero, J.S. (1996) Function–behavior–structure paths and their role in analogy-based design, *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, Vol. 10 No. 04, pp. 289–312.
Rajamani, R. (2012) Roll Dynamics and Rollover Prevention, in Ling, F.F. (Ed.),*Vehicle Dynamics and Control*, Springer US, Boston, MA, pp. 427–456.
Regli, W.C., Hu, X., Atwood, M. and Sun, W. (2000) A survey of design rationale systems:

approaches, representation, capture and retrieval, *Engineering with computers*, Springer, Vol. 16 No. 3, pp. 209–235.
Richardson III, J.L., Summers, J D and Mocko, G M. (2011) Function Representations in Morphological Charts: An Experimental Study on Variety and Novelty on Means Generated, *Interdisciplinary Design: Proceedings of the 21st CIRP Design Conference*, p. 76.
Rodenacker, W. (1971) *Methodisches Konstruieren*, Springer Verlag, Berlin, Germany.
Sasajima, M., Kitamura, Y., Ikeda, M. and Mizoguchi, R. (1995) FBRL: A Function and Behavior Representation Language, *International Joint Conference on Artificial Intelligence*, Montreal, Canada.
Schultz, J., Sen, C, Caldwell, B., Mathieson, J., Summers, J D and Mocko, G M. (2010) Limitations to Function Structures: A Case Study in Morphing Airfoil Design, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Montreal, Canada, pp. DETC2010–28559.
V. Sembugamoorthy, B. Chandrasekaran, Sembugamoorthy, V. and Chandrasekaran, B. (1986) Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems, in Kolodner, J. and Riesbeck, C.K. (Eds.),*Experience, Memory, and Reasoning*, Erlbaum, Hillsdale, NJ, pp. 47–53.
Sen, C, Summers, J D and Mocko, G M. (2011) Exploring potentials for conservational reasoning using topologic rules of function structure graphs, *Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 9*, pp. 377–388.
Sen, C. and Summers, J. D. (2012) A Pilot Protocol Study on How Designers Construct Function Structures in Novel Design, in Gero, J. (Ed.),*5th International Conference on Design Computing and Cognition*, College Station, TX, p. No. 37.
Sen, C., Summers, J. D. and Mocko, Gregory M. (2011) A protocol to formalise function verbs to support conservation-based model checking, *Journal of Engineering Design*, Taylor & Francis, Vol. 22 No. 11-12, pp. 765–788.
Sen, C., Summers, J. D. and Mocko, Gregory M. (2012) A Formal Representation of Function Structure Graphs for Computer-Directed Modeling and Conservation-Based Reasoning, *ASME Transactions Journal of Computing and Information Science in Engineering*, Vol. accepted.
Smith, G., Richardson, J., Summers, J D and Mocko, G M. (2012) Concept exploration through morphological charts: an experimental study, *Journal of mechanical design*, American Society of Mechanical Engineers, Vol. 134 No. 5, pp. 051004.1–10.
Srinivasan, V., Chakrabarti, A. and Lindemann, U. (2012) A Framework for Describing Functions in Design, *International Design Conferences*, Dubrovnik, Croatia, pp. 1111–1122.
Suh, N. (1999) A Theory of Complexity, Periodicity, and the Design Axioms, *Research in Engineering Design*, Vol. 11, pp. 116–131.
Summers, J D and Shah, J.J. (2004) Representation in engineering design: a framework for classification, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Salt Lake, UT, p. DTM–57514.
Ullman, D.G. (2010) *The Mechanical Design Process*, McGraw-Hill, New York, NY.
Ullman, D.G., Dietterich, T.G. and Stauffer, L.A. (1988) A model of the mechanical design process based on empirical data, *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, Vol. 2 No. 01, pp. 33–52. Retrievedfrom http://journals.cambridge.org/abstract_S0890060400000536
Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. and Tomiyama, Tetsuo. (1996) Supporting Conceptual Design Based on the Function-Behavior-State Modeler, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 10 No. 4, pp. 275–288.
Vermaas, P.E. (2013) On the Co-Existence of Engineering Meanings of Function: Four Responses and Their Methodological Implications, *Artificial Intelligence in Engineering Design Analysis and Manufacturing*, Vol. 27 No. 3, pp. 1–19.
Vescovi, M., Iwasaki, Y., Fikes, R. and Chandrasekaran, B. (1993) CFRL: A Language for Specifying the Causal Functionality of Engineered Devices, *11th National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Washington, DC.
Winston, P.H. (2005) *Artificial intelligence*, *Reading, Mass.: Ad-dison-Wesley*, Addison-Wesley, Reading, MA, 4thed.
Yang, S., Patil, L. and Dutta, D. (2010) Function Semantic Representation (FSR): A Rule-Based Ontology for Product Functions, *ASME Transactions Journal of Computing and Information Science in Engineering*, Vol. 10, p. 031001.