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

MECHANICAL ENGINEERING MODELLING LANGUAGE (MEML): REQUIREMENTS FOR CONCEPTUAL DESIGN

Joshua David SUMMERS (1), David ROSEN (2)
1: Clemson University, United States of America; 2: Georgia Tech University, United States of America

ABSTRACT

The objective of this paper is to define the characteristics for a new approach for capturing, representing, and modelling mechanical engineering knowledge in a Mechanical Engineering modelling language (MEml) that was discussed at a recent NSF sponsored workshop. With a focus on conceptual design, the paper provides a vision of design product modelling, relating research on function representations to work on ontologies, reasoning methods, and similar work in other domains. A series of three function-based representations is explored through an application to conceptual design of an additive manufacturing process. A comparison is offered of the types of information represented and reasoning supported by these representations, in light of the application. Finally, gaps are highlighted between the capabilities of these representations and the requirements for MEml to support conceptual design.

Keywords: function, requirements, conceptual design, modeling, representation, reasoning

Contact:
Prof. Joshua David Summers
Clemson University
Mechanical Engineering
Clemson
29634-0921
United States of America
jsummer@clemson.edu

1 MEML: FOUNDATIONS

The objective of this paper is to define the characteristics for a new approach for *capturing*, *representing*, and *modeling* mechanical engineering knowledge (MEml: Mechanical Engineering modeling language). The goal is to capture and define knowledge about ME principles, such as mass and energy conservation laws, and associated the artifacts considering different modeling points of view, such as free body diagrams and system behavior models. Further, the knowledge about *how to formulate* models of devices and design decisions should be captured, yet this design process modeling language is out of scope for this paper. A formalized language of ME is envisioned to allow engineers to communicate precisely between each other and with computers through extended mixed initiative reasoning systems. The resulting formalized MEml should be implemented in an open-knowledge repository which can then serve as a key component to augment engineering design research and education, in addition to off-loading routine engineering activities in practice.

When considering the challenges associated with developing a true ME modeling language (MEml), collaboration with many disciplines is necessary. A convergence of artificial intelligence, engineering informatics, description logics, and the semantic web with ME design research is an enabling factor to realize the vision. This convergence was initiated in 2012 with an NSF sponsored workshop, the preliminary results being presented in (Rosen and J D Summers, 2012). This paper, continuing in this direction, focuses on the information that should be captured and how these information elements relate to different types of reasoning activities for conceptual design.

2 DESIGN PRODUCT MODELLING

The Vision

There are several different approaches and views of how to model the information used to define and design products. These approaches have different objectives, such as the archival and standards approach inherent to the core product model (Fenves et al., 2008), to the design activity elicitation goals of Function-Behavior-Structure (FBS) models (Gero and Kannengiesser, 2004), to supporting specific types of reasoning, such as analogical reasoning of the Structure-Behavior-Function model (Bhatta et al., 1994), to tracing design problem and solution evolution of the P-Map (Dinar et al., 2011), or to formalization of single engineering domains for early stage reasoning with function structures (Hirtz et al., 2002; Sen et al., 2011). The emphasis on much of the past research has centered on function and behavior modeling in addition to linking this information to structural information, such as components, parameters, or properties.

In 20 years, the authors anticipate that the semantic web will be an integral part of engineering design and engineers will be able to connect to all publicly available design knowledge. The publicly available design knowledge, found online, will be indexed and available for search in a variety of representations, such as text based, drawings, and mathematical models. Design tools will capture and store the semantics of the design in an open ontology format. Due to the integration of design knowledge from across disciplines the boundaries between domains while designing new products will blur. Engineering design analysis will become much more powerful and provide a holistic view of design that will facilitate better innovative designs. The semantic web and associated technologies will enable engineering mathematical models to be automatically constructed from first principles and the context of the problem domain, automatically solved and validated, and automatically interpreted to support mechanical engineering design. Based on these assertions, dramatic changes are needed to enable engineers to economically capture their design knowledge, such that trade-offs between knowledge capture and cost will be minimized.

Requirements

To achieve the vision, significant research will be needed for many years. For conceptual design, research related to MEml is needed in design product modeling, reasoning methods to generate and evaluate design concepts, and representation schemes for the wide variety of concept design information. Specifically, we focus on five types of reasoning. **Analogical reasoning** is valuable for generating design concepts, **morphological analysis** is needed to generate system concepts from subsystem concepts, **behavior prediction** and **failure identification** are necessary to evaluate designs, and **archiving** stores designs for communication and re-use.

Several types of information are commonly recognized as important in the design process, especially conceptual design: requirements, functions, behaviors, working principles, parameters, mathematical expressions, and structure or geometry. While there is not complete agreement (Eckert et al., 2011), a few definitions are offered here. Requirements capture the intent, purpose, and objectives associated with the design problem. Functions are the actions for the device, be that acting on the environment, on other devices or artifacts, or acting on the user, or they may be actions of the user realized through the device. Behaviors are the reactions to external stimuli and are realized through the working principles. Parameters are the dimensions, variables and values, or other direct and indirect descriptions of the physical system, user, or environment. Math expressions relate parameters through algebraic or differential relationships. Finally, structure or geometry is used to define the components, sub-assemblies, and assemblies that are being designed.

Foundations

Transformative research is needed to bring together the domains and associated computational tools for mathematics, physics, mechanics, and ME design. Some foundational work is sampled here that represents successes in large ontology and reasoning system development in the math, biological, and engineering domains. Significant work has been done in the mathematics community to formally define the standard syntax and even logic of mathematical equations, resulting in representations such as MathML 3.0 (a W3C recommendation) and the OpenMath standard by (OpenMath.org). The level of semantic formalism in the mathematical domain has reached the point where proofs can be checked automatically by machines and theorems can be proven automatically. However, while these standards rigorously define syntax and logical structures of mathematical equations, they do not semantically define their application context, such as the physical phenomenon or the underlying physical principles that the equations may represent. In order for tools to be capable of automated reasoning about engineering models that are employed and for these models to be easily shared and reused, their application contexts must be formerly defined. This means that engineering domain ontologies are needed to enrich mathematical models with the semantic context of engineering problems.

The biological and biomedical communities have demonstrated this. These communities have developed a number of well curated and open domain ontologies (such as the Gene Ontology, the Biological Process Ontology, the Chemical Entities of Biological Interest Ontology, and the Foundational Model of Anatomy Ontology (Bailey, 2008; Hong et al., 2008; Sehgal et al., 2011)), community-standard mathematical markup languages for representing models involved in system biology, such as SBML, and even a web repository of over 400 mathematical models (Kieffer et al., 2008; Lange, 2012). As a result, numerous tools have been created for these communities. For example, over 100 freely downloadable tools have been developed for the Gene Ontology alone (Khatri and Drăghici, 2005). Recently, researchers have leveraged the existing suite of open domain ontologies to integrate mathematical system biology models with appropriate domain ontologies, providing the context upon which the models are based and enabling various automated tasks to be executed, such as model consistency verification.

In the engineering community researchers have produced a number of domain ontologies, such as engineering mathematics, physical phenomenon, product modeling, functional modeling, engineering analysis models, design optimization, and so on. However, the representation, development and management of ontologies in the engineering domain is largely ad hoc, the result of small research groups producing domain ontologies in isolation which are often not opened to the Semantic Web nor well curated. Large domain gaps still exist. For example, little work has been done to semantically describe engineering materials, continuum mechanics, control theory, many design innovation methods, conservation principles, etc. The engineering community needs a semantic engineering modelling language that is built on an existing semantic mathematical markup language, such as OMDoc, or a new mathematical markup language extended with application-specific information as a mechanism for publishing, sharing, and archiving engineering mathematical models. Progress has been reported on using SysML to do this for modelling systems engineering problems (Kerzhner and Paredis, 2011). Finally, tools are needed that support engineers as they develop engineering models, so engineers can work at the level of abstraction most suitable for their field. Such tools should be able to guide engineers to ensure that models developed are consistent with fundamental principles, such as conservation of mass and energy, and conform to domain constraints. MEml should provide the vocabulary, grammar, and expressiveness for such tools to be developed.

3 EXAMPLE DESIGN CHALLENGE: LAYERED MANUFACTURING

In this and subsequent sections, several knowledge modelling approaches will be applied to modelling of a class of manufacturing processes so that the approaches can be compared and contrasted. Additive manufacturing (AM) refers to the use of layer-based additive processes to manufacture finished parts by stacking layers of thin 2-D cross-sectional slices of materials (Gibson et al., 2009). These processes allow fabrication of parts with high geometric complexity, material grading, and customizability. Typical AM processes fabricate layers by either patterning material (depositing and processing) directly or by patterning energy onto a bulk material, such as a powder bed. Fused-deposition modelling (FDM) is an example of the first type, where filaments of material are extruded in a pattern that creates a part cross-section, while selective laser sintering (SLS) is an example of the latter. In SLS, a laser scans the top surface of a powder vat and melts the powder so that a solid part cross-section is formed. A schematic of the process is shown in Figure 1, along with a representative function structure. After a part cross-section is formed, the recoat blade sweeps a new powder layer over the build area and the process repeats.

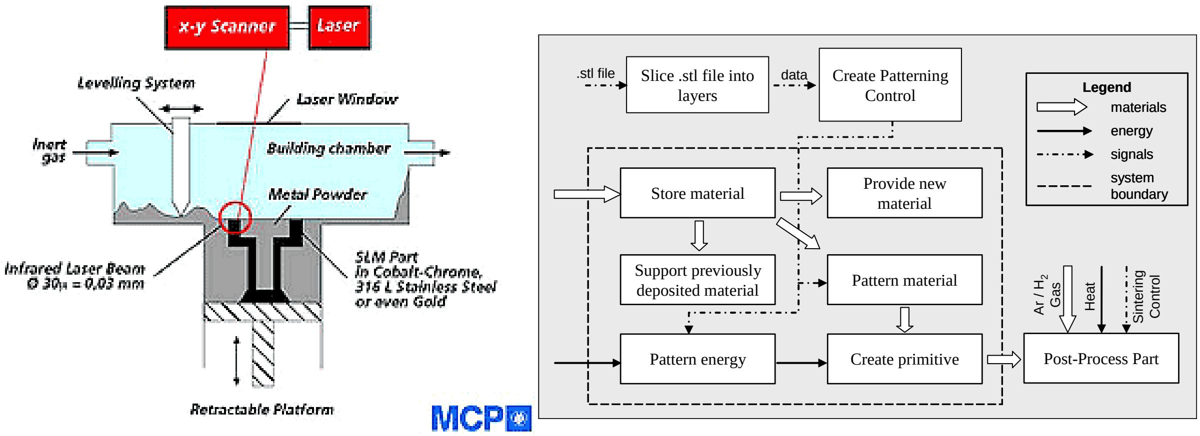

Figure 1: Selective Laser Sintering Process

Function Structure

The first modeling approach is the function structure for transformations of materials, energy, and signals (Hirtz et al., 2002; Pahl et al., 2007; Sen et al., 2011). This approach is limited to function, while the next two include function, behavior, and structure. Even with this high level abstraction with current extensions and refined vocabularies and grammatical rules for model construction, significant qualitative physics reasoning can be achieved. The first principles and transport phenomena laws of conservation and the efficiency loss guarantee of the second law of thermodynamics can be determined based on graph analysis. For instance, any energy that passes across a system boundary must be recovered outside of the boundary or be converted into work. Moreover, as every transformation will require some energy and any energy conversion will result is a non-recoverable energy loss, the function

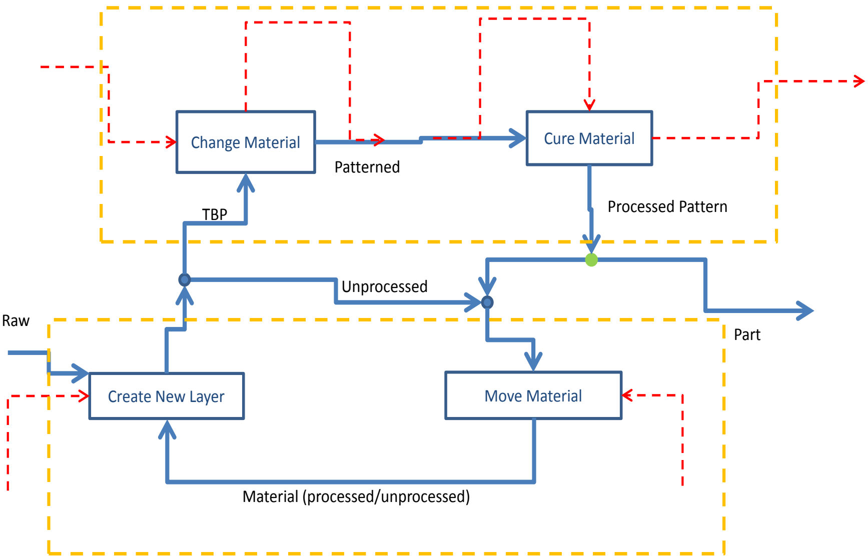

Figure 2: Function Structure to Describe Layered Manufacturing

structure can be analyzed for physics based realism. Figure 2 illustrates a possible function structure created to describe a layered manufacturing process.

Note that this is a generalization of the function structure in Figure 1b. It is also a slight modification since it models an additive manufacturing process that patterns material, not energy as in SLS.Each of these functions can then be realized through different working principles or concept fragments, as one would find in a morphological matrix. In this manner, the function structure can be restricted to allow only high level feasible solutions while providing the skeleton on which to start to combine working principles into an embodied solution. In the case of AM, raw material is input into the system and a defined part is output that consists of layers comprised of processed cross-section patterns. In this solution architecture, it is assumed that all of the input raw material is converted into the part, but this assumption can be relaxed. Additionally, energy is input into the system through the function of changing material, by energizing the material either by melting it, raising the temperature, or vibrating the material (Sen et al., 2011). This energized material is then de-energized by curing the materials or other working principles.

Function-Behavior-Structure (FBS)

After reasoning at the function level of abstraction, it is natural to progress to the next level of detail. A different approach to modelling the design solution and process is through the Function-Behavior-Structure model (Gero and Kannengiesser, 2004). Within this modelling scheme, the engineer progresses from requirements to functions to expected behaviour to actual behaviour to structure. Several patterns and prototype templates have been identified to explain different design activities. An advantage of this approach is that the structure of the solution can be directly linked to the initial requirements or functions through the behaviours. A simplified FBS model for the layered manufacturing system is proposed in Figure 3.

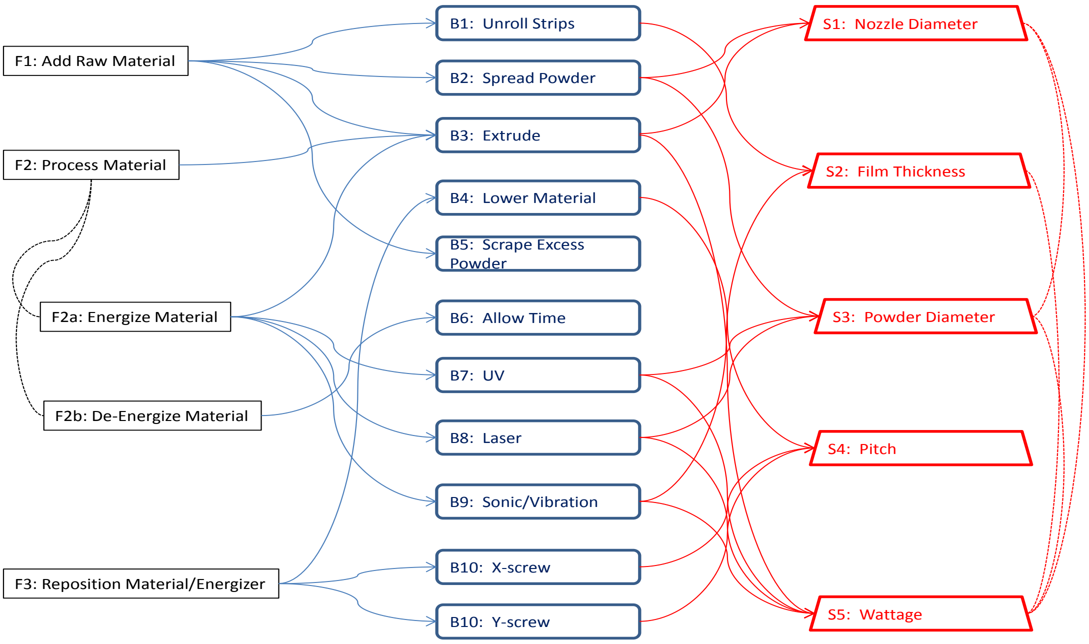

Figure 3: FBS Model of General Layered Manufacturing System Options

This is the evolved model with the connections and relations between the function elements (e.g., add raw material), the behaviours (e.g., unroll strip), and the structure (e.g., strip thickness). In this case, the function goal of adding raw material to be processed into a new shape can be realized through a behaviour of unrolling a strip (such as through ultra-sonic consolidation). There are other optional behaviors that can be mapped to this function, such as spreading powder (SLS) or extrusion (FDM). Therefore, in this model, the behaviours are not all required. Either multiple distinct FBS models can be generated or the combined options can be modelled. As multiple different possible behaviours are developed, structural solutions can be proposed. These might be parameters or entire components. The parameters might relate to and influence many different behaviours. Moreover, there might be direct relationships between the parameters. A limitation to this modelling scheme is that there is not currently an available controlled vocabulary to refine and limit the modelling scope. Thus, designers can create models of similar products and devices, but with entirely differently defined parameters, behaviours, or functions. However, the design patterns that can be traced through the evolution of the FBS models can be useful for identifying design justification and rationale. Moreover, the types of information captured here are widely recognized as critical engineering information domains by others using different terminology and resolution (Pahl et al., 2007; Suh, 1990). While automated qualitative physics and behavioural simulation is not explicitly supported through the modelling approach, these can be realized through a controlled vocabulary achievable within restricted application domains.

Structure-Behavior-Function Representation

The third approach adds additional information to the model. Over the past 20 years, Ashok Goel has proposed the SBF model for devices and their operation. Device structure consists of components and substances, where substances are materials, energy, or signals that flow through the components and have behavioural properties (e.g., density, melting temperature) and values. Behaviour is represented by a series of states and transitions between these states, where a state represents the properties of a substance at a specific location or time. State transitions are annotated with causal, structural, and functional contexts in which the transitions occur. Function and behaviour are specified hierarchically so that in describing the behaviour of an overall device, internal behaviours reference functions that are performed by components in the device, and so on. SBF models of op-amps and gyroscope follow-on mechanisms, among others, have been demonstrated to support case-based and analogical reasoning. By learning about feedback control using an op-amp and inverting amplifier, these researchers demonstrated that the concept of feedback control can be applied to other engineering systems to improve their performance, in this case a gyroscope follow-on mechanism.

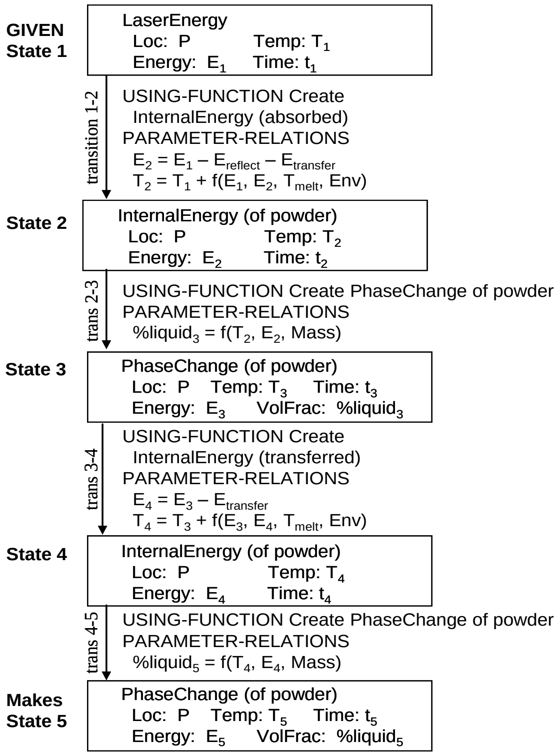

Figure 4: Behavior Model of SLS Powder Processing

As presented, the behaviour models are detailed and complete. A candidate model for the laser processing in an SLS machine, corresponding to the "Create primitive" function is shown in Figure 4. The scanning laser beam passes a point on the powder surface, P, heats it up, and causes it to melt, assuming that enough laser energy was received at point P. After the laser passes, the melted powder cools due to conduction and convection and solidifies into a frozen shape. Parametric equations are included that enable quantitative reasoning, if desired. Such a detailed behaviour model is needed to support simulation, explanation, and evaluation. The model makes causality explicit and has references to components, their functions, and to relevant physical principles. However, the model may be more detailed than needed in order to support design processes for new or alternative AM processes. From a different perspective, the construction of such models could be partially automated if a knowledge base had models of physical phenomena and the capability of applying them to components. For example, if a model of the solid material irradiation physical phenomenon was available, then the SBF model constructor would not have to specify states 2 or 3 and their transitions, since the effects of irradiating a solid were already available. The reasoning system need only instantiate the physical phenomenon in the context of the laser and powder bed.

4 COMPARISON OF INFORMATION CAPTURED

When comparing the function-based approaches (Table 1), there are different types of information captured, in addition to the vocabulary of the information and the grammar of the representation (J D

Summers and J.J. Shah, 2004). This comparison is provided for the three types of models described above, including two variations on function structures.

Table 1: Representational Comparison of ME Conceptual Design Modeling Approaches

			Function Structures (Pahl et al., 2007)	**Formalized Function Structures** (Sen et al., 2011)	**FBS** (Gero and Kannengiesser, 2004)	**SBF** (Bhatta et al., 1994)
Vocabulary	Type		Object-Relation	Object-Relation-Modifier	Object-Relation	Object-Relation-Modifier
Vocabulary	Size		53 Functions + 45 Flows(Hirtz et al., 2002)	16 atomics (functions, relations, attributes)	F, Be, Bs, S	S, B, F, param
Vocabulary	Flexibility		User defined	Pre-defined (atomics), Pre-defined (templates), User defined (templates)	User defined	User defined
Vocabulary	Information	Requirements			X	
Vocabulary	Information	Function	X	X	X	X
Vocabulary	Information	Behavior		Some	X	X
Vocabulary	Information	Working Principles			X	X
Vocabulary	Information	Parameters		Some	Some	X
Vocabulary	Information	Structure				
Grammar	Local		None	Yes	None	None
Grammar	Global		None	Yes (physics based)	None	None
Grammar	Validation		Manual	In-Process, Post-Hoc	Manual	Post-Hoc
Expression			Iconic	Iconic, Computational	Iconic, Logical	Iconic, Mathematical, Computational
Purpose			Synthesis	Synthesis, Communication (archival), Analysis	Synthesis	Synthesis, Communication (archival), Analysis
Abstraction			High	High	High	Moderate

Specifically, the vocabulary is characterized by the types of elements involved (object-relation, object-relation-modifier, non-defined), the size or number of classes available in the vocabulary, and the flexibility or rigidity of definition of the vocabulary. The vocabulary can also be examined with respect to the engineering information domains which are captured. Considering a traditional prescriptive model of engineering design (Pahl et al., 2007), information types including requirements, function, behavior, working principles, parameters, and geometry are considered in this comparison. The grammar, or structure, of the representation includes characterization in terms of local rules and constructs, global rules and constructs, and how the models are validated (manual, automated post-hoc, and in-process). Three additional characteristics of engineering representations deal with the type of expression (textual, graphical, mathematical, or computational), the intended purpose of the representation (synthesis, communication, and analysis), and the level of abstraction or the degree of inferencing required to extract knowledge from the models.

All four representations have explicit models of function, meaning the desired behavior of the device to be designed. They differ greatly in their capability of representing and reasoning with behavior. This is due in large part to their intended usages, where the function structure based representations do not have powerful behavior models, but the FBS and SBF representations do. SBF has the best capabilities for representing and reasoning with behavior. However, even it falls short of strong qualitative physics reasoning capabilities. Physical principles are not explicit in any of the representations, which are essential for providing models to support physics-based reasoning.

The authors hypothesize that the engineering and physical semantics of mathematical expressions must be represented explicitly in MEml in order to support mathematical reasoning and meta-reasoning. However, none of the representations has rich information models of mathematical expressions and few attempts have been made in associating physical semantics with math models in the engineering literature. This comparison is provided as an illustration of how different representations can be compared systematically to extract potential opportunities for evolution. If MEml is to be realized, the information that is represented within should be capable of mapping to these, and other, conceptual design modeling approaches.

5 COMPARISON OF REASONING SUPPORTED

While representation provides the framework to capture engineering knowledge, this knowledge is of little value unless supported by reasoning strategies and tools. Early optimistic efforts included an inferencing system aimed at mimicking human designers (Dixon et al., 1987; Howe et al., 1986; Orelup et al., 1987) and the concept of a design compiler (Ward, 2001). Current engineering reasoning systems can be classified by whether the representation uses functions or predicates as the primary means of relating objects to each other. Thus, in addition to comparing the modeling approaches from a representational point of view, the reasoning activities that are supported with the representations should be considered. As a reminder, five activities under consideration: archiving, analogical reasoning, morphological analysis, behavior prediction, or failure identification.

Archiving is identified to support communication between and within designers and across time. For instance, an engineer may want to share ideas and her understanding of a problem with other team members. Therefore, a common semantically understood representation is needed. This same engineer also may want to revisit the model at a later date for review. The level of formalism of the representation determines the degree to which the model can be shared with others with a common understanding or how easily models can be queried and found from databases. Databases are found for each of the modeling approaches (Bohm et al., 2008; Dai et al., 1996; Wiltgen et al., 2011); however, none of the approaches supports meta-information about contexts that would be useful for adaptation and reuse for other engineering problems.

Analogical reasoning is important in developing novel designs with improved functionality. It is a process of mapping from one information domain to another and back, bringing with it possible behaviors and working principles that could achieve the desired goals. Analogical reasoning is supported by each modeling approach (A.K. Goel et al., 1997; Linsey et al., 2008; Qian and Gero, 1996). The SBF representation has the most advanced analogical reasoning capability since it has explicit causal relationships in its behavior model. It is important that MEml contain such causal relationships, which may be automatically identified if rich physical representations are available.

Morphological analysis is similar to aspects of analogical reasoning, in that morphological analysis is a mapping from one information domain to another to explore possible approaches to realizing the requirements. For instance, the function of converting rotational motion to translation motion can be achieved through different working principles such as screws, slider cranks, or pulley lifts. Morphological analysis has been shown to be supported by each of the modeling approaches or variations (Gero et al., 2012; Richardson III et al., 2011; Vargas-Hernandez and J.J. Shah, 2004).

In order to support the analogies beyond explicitly defined concept mappings, a level of *behavior predictions* should also be supported. These behavioral predictions could be at the qualitative or quantitative physics level. Should the behavioral prediction be supported, *failure identification* can be supported to identify sensitivities and potential areas of the device that might prevent satisfaction of the requirements under different scenarios. Finally, morphological analysis is a reasoning activity that relates to the analogical reasoning. These activities are also supported, to varying degrees, by the modeling approaches (Hamraz et al., 2012; Sen et al., 2011; R.B. Stone et al., 2005).

6 GAPS AND RECOMMENDATIONS

If MEml is to become the underlying modeling language to support engineering design, then it should be capable of translating into each of these three popular modeling approaches while supporting the reasoning that is achieved through them. This means that the representation of MEml should span multiple engineering information domains (requirements, functions, behavior, working principles, parameters, and structure), providing links between these. The information that is common to all of these modeling approaches is the function. However, the function vocabularies vary and are not currently standardized. Therefore, it is first necessary to determine whether a function that is described for function structures (Hirtz et al., 2002; Sen et al., 2011), also captures the same necessary information as the functions of SBF and FBS. This requires a deeper study of the definition of functions, their role in design, and how these relate to the other domains. A formal benchmarking of the existing models is needed to compare them in terms of representation and reasoning. This paper provides a first step towards this end. It is not clear that one of these different representations is more critical than the others; rather it appears that each approach has relative strengths. Therefore, MEml should be able to transition between these. Beyond comparison, it is necessary to demonstrate the transferability between these modeling approaches.

The general strength of the function structure approach seems to be the formalism potential in defining a controlled vocabulary. With this vocabulary, generic, physics based reasoning can be supported. However, the representation deals with only one domain. The FBS approach helps to highlight the morphological space as the solution parameters can map back through behavior to the functions or goals. This allows for a more direct connection of the design problem to the design solution. Finally, the SBF modeling approach appears to be the richest of the three in terms of parametrically describing the states of operation of a device. This richness supports more detailed analysis of the potential solutions and a more systematic approach to determining appropriate levels of resolution or abstraction for models. These strengths should be realized and integrated together in the MEml.

ACKNOWLEDGMENTS

We gratefully acknowledge the support from NSF (#1136452). The views and opinions expressed here are of the authors exclusively.

REFERENCES

Bailey, R. (2008) Comparative Study of Undergraduate and Practicing Engineer Knowledge of the Roles of Problem Definition and Idea Generation in Design, *International Journal of Engineering Education*, Vol. 24 No. 2, pp. 226–233.

Bhatta, S.R., Goel, A.K. and Prabhakar, S. (1994) Innovation in Analogical Design: A Model Based Approach, *Artificial Intelligence in Design*, Dordrecht, The Netherlands.

Bohm, M.R., Stone, R.B., Simpson, T.W. and Steva, E.D. (2008) Introduction of a data schema to support a design repository, *Computer-Aided Design*, Elsevier, Vol. 40 No. 7, pp. 801–811.

Dai, W., Zhang, D. and Crawford, J. (1996) Intelligent management of building-product database, *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, IEEE, Vol. 2, pp. 1563–7.

Dinar, M., Shah, J., Hunt, G., Campana, E. and Langley, P. (2011) Towards a formal representation model of problem formulation in design, *Proceedings of ASME DETC*.

Dixon, J., Howe, A., Cohen, P. and Simmons, M. (1987) Dominic I: Progress Toward Domain Independence in Design by Iterative Redesign, *Engineering with Computers*, Vol. 2, pp. 137–145.

Eckert, C., Alink, T., Ruckpaul, A. and Albers, A. (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.

Fenves, S.J., Foufou, S., Bock, C. and Sriram, R.D. (2008) CPM2: A Core Model for Product Data, *Journal of Computing and Information Science in Engineering*, Vol. 8, p. 14501.

Gero, J.S., Jiang, H. and Williams, C.B. (2012) Does Using Different Concept Generation Techniques Change the Design Cognition of Design Students?, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, Chicago, IL, p. #71165.

Gero, J.S. and Kannengiesser, U. (2004) The situated function–behaviour–structure framework, *Design Studies*, Vol. 25 No. 4, pp. 373–391. doi:10.1016/j.destud.2003.10.010

Gibson, I., Rosen, D.W. and Stucker, B. (2009) *Additive manufacturing technologies: rapid prototyping to direct digital manufacturing*, Springer.

Goel, A.K., Bhatta, S.R. and Stroulia, E. (1997) KRITIK: An Early Case-Based Design System, in Maher, M. Lou and Pu, P. (Eds.),*Issues and Applications of Case-Based Reasoning in Design*, Erlbaum, Mahwah, NJ, pp. 87–132.

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.

Hong, E.L., Balakrishnan, R., Dong, Q., Christie, K.R., Park, J., Binkley, G., Costanzo, M.C., et al. (2008) Gene Ontology annotations at SGD: new data sources and annotation methods., *Nucleic acids research*, Vol. 36 No. Database issue, pp. D577–81.

Howe, A., Cohen, P., Dixon, J. and Simmons, M. (1986) Dominic: A Domain-Independent Program for Mechanical Engineering Design, *Artificial Intelligence in Engineering Design*, Vol. 1, pp. 23–28.
Kerzhner, A. and Paredis, C. (2011) Model-Based System Verification: A Formal Framework for Relating Analyses, Requirements, and Tests, in Dingel, J. and Solberg, A. (Eds.),*Models in Software Engineering*, Springer Berlin Heidelberg, Berlin, Heidelberg, 6627/2011ed., Vol. 6627, pp. 279–292.
Khatri, P. and Drăghici, S. (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems., *Bioinformatics (Oxford, England)*, Vol. 21 No. 18, pp. 3587–95.
Kieffer, S., Avigad, J. and Friedman, H. (2008) A language for mathematical knowledge management, Logic in Computer Science, , pp. 51–66. Retrievedfrom http://arxiv.org/abs/0805.1386
Lange, C. (2012) Ontologies and languages for representing mathematical knowledge on the Semantic Web, *Semantic Web*, Vol. in press, pp. 1–40.
Linsey, J.S., Wood, K.L. and Markman, A.B. (2008) Modality and representation in analogy, *AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing)*, Cambridge Univ Press, Vol. 22 No. 2, p. 85.
Orelup, M., Dixon, J. and Simmons, M. (1987) Dominic II: More Progress Towards Domain Independent Design by Iterative Redesign, *Winter Annual Meeting*, Boston, MA.
Pahl, G., Beitz, W., Wallace, K. and Blessing, L. (2007) *Engineering Design: A Systematic Approach*, Springer-Verlag London Limited, London, 3rded.
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. Retrievedfrom http://journals.cambridge.org/abstract_S0890060400001633
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.
Rosen, D.W. and Summers, J D. (2012) Mechanical Engineering Modeling Language (MEml): Necessary Research Directions, *International Conference on Innovative Design and Manufacturing*, Taipei, Taiwan.
Sehgal, A.K., Das, S., Noto, K., Saier, M.H. and Elkan, C. (2011) Identifying relevant data for a biological database: handcrafted rules versus machine learning., *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, Vol. 8 No. 3, pp. 851–7.
Sen, C., Summers, J. and Mocko, G. (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. doi:10.1080/09544828.2011.603295
Stone, R., Tumer, I. and Stock, M. (2005) Linking product functionality to historic failures to improve failure analysis in design, *Research in Engineering Design*, Springer, Vol. 16 No. 1, pp. 96–108.
Suh, N. (1990) *The principles of design*, Oxford University Press, New York, NY, Vol. 990.
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.
Vargas-Hernandez, N. and Shah, J.J. (2004) 2 nd-CAD: a tool for conceptual systems design in electromechanical domain., *Journal of Computing and Information Science in Engineering(Transactions of the ASME)*, Vol. 4 No. 1, pp. 28–36.
Ward, A. (2001) Mechanical Design Compilers, in Antonsson, E. and Cagan, J. (Eds.),*Formal Engineering Design Synthesis*, Cambridge University Press, Cambridge, UK, pp. 428–441.
Wiltgen, B., Goel, A. and Vattam, S. (2011) Representation, indexing, and retrieval of biological cases for biologically inspired design, *Case-Based Reasoning Research and Development*, Springer, pp. 334–347.