CHALLENGES IN DEVELOPING AN ONTOLOGY FOR PROBLEM FORMULATION IN DESIGN

Dinar, Mahmoud; Park, Yong-Seok; Shah, Jami J
Arizona State University, United States of America

Abstract

The need for capturing and documenting problem formulation data exists for early stages of conceptual design. In this paper, we review two versions of our Problem Map ontological framework which we have used for expressing problem formulation data of different designers with the intent of understanding differences among them. We discuss some of the challenges that we have faced in developing and using the ontology for the annotation of problem formulation data for both introspective and reflective annotators. They include unclear boundaries of definitions for a few entities; difficulty in striking the right balance between a light, structured, easy to express ontology, and a fluid ontology with many entities that increases the chance of expressing data fragments but lowers accuracy among annotators; and making the ontology easier to learn for users, both introspective and reflective annotators. We give examples of how the ontology represents problem formulation data fragments, and examples of challenges from collected data.

Keywords: Ontologies, Conceptual design, Design informatics

Contact:
Mahmoud Dinar
Arizona State University
Mechanical and Aerospace Engineering
United States of America
mdinar@asu.edu

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

In research in design thinking, problem formulation has been the latest of the subjects to catch on with the processes that usually follow it. One reason is that some have equated problem definition solely with the methods that go under the label of requirement engineering. Even though the notion of co-evolution of problem and solution spaces has been accepted (Dorst and Cross, 2001; Maher and Tang, 2003; Maher et al., 1996; Smulders et al., 2009), the problem formulation aspect still lacks dedicated research efforts. We have, therefore, defined our primary research goal as understanding the elements and the processes that relate to problem formulation, and how they affect creativity. To reach this goal the following roadmap was outlined:

1. Observe the process in action.
2. Choose an appropriate representation for modelling the activity.
3. Create a modelling framework and model captured data.
4. Search for patterns that reveal differences between more creative and less creative designers.
5. Describe causal relationships; propose and test hypotheses/theories.

Our first two papers (Danielescu et al., 2012; Dinar et al., 2011) covered the first two steps. We collected two sets of protocols for two different design problems from a few designers to see what representation and level of detail could help us identify differences among designers. Inspired by Concept maps (Novak and Gowin, 1984), which have been used in education research for knowledge representation and comparison to normative models of expert knowledge, we created the first Problem Maps (Dinar et al., 2011). The next section will describe this representation. Later in the paper, we will also discuss the shortcomings of Concept maps that led us to add more specificity (with a limited set of labels to choose for coding data fragments) and structure in our modelling framework.
To improve expressiveness of the model, based on what was important but missing in the initial version, an updated ontology emerged. This will be discussed in section 3. The ongoing challenges in employing our ontologies to express or code problem formulation data will be discussed further in section 4. The paper follows with a review of other pertinent representation models and work on ontologies in conceptual design, and finishes with concluding remarks.
Before we begin the review of the ontologies, we should explain why they matter to us. Ontologies are important not merely because of their application in knowledge representation, but because they intimately involve language. The growth of natural language understanding and text mining methods has raised the potential in understanding and aiding conceptual design when ideas are best expressed with words. Even though sketching is supposed to be more effective in expression (Larkin and Simon, 1987), in early stages of problem formulation, prior to expressing any forms or embodiments, words can have a higher efficiency of describing abstract design thoughts (Doumont, 2002; Willows, 1978).
Another issue that involves ontologies is search through words. Regardless of growing computing power, search results can become overwhelming for the user to filter through when employing knowledge bases without a proper structure that maps onto the domain at hand. Most knowledge bases such as WordNet (Miller, 1995) have ontologies more suited towards common sense knowledge, not design or engineering. There is a need for an ontology specific to design but also not limited to technical terms which can be found in some design repositories such as Bohm et al. (2008), since the fuzzy front end of the early stages of conceptual design, especially for novel designs, is often described in a less formal language. The intent of this paper is not to compare this ontology to others. It is to stir a debate on the importance and the application of ontologies in early conceptual design, and the challenges that we have faced in developing and employing an ontology for expressing problem formulation.

2 REVIEW OF THE INITIAL PROBLEM MAPS ONTOLOGY

2.1 Elements of Problem Maps

Problem Maps are a representation of different aspects of the design problem that designers think about when formulating a design problem. Our first attempt was based on a protocol study where we asked two groups of novice students and an expert designer to design a remotely-controlled model plane for a competition with different scoring weights for speed of the plane and its load carrying capacity in multiple missions. The exploratory protocol study resulted in creating a visual representation of data fragments and the relations expressed among them in a state model that facilitated a comparison of different designers' thinking process, and their progression. Here, the ontology structure is in the name of the entity types, e.g., *Component*. It should be mentioned that the co-evolution of problem and solution spaces (Dorst and Cross, 2001) implies that elements such as components will be present in expressing problem formulation. The collection of instances under various entities along a timeline formed the Problem Map. Figure 1 shows an end state (after designers finished working about an hour on the problem) for one novice group and the expert. In order to make the comparison easier, similar fragments were given the same label in the model. Even though this naming and categorization was done by a [coder] researcher, the expression of design thoughts should be the main focus. This can be taken as a process of *annotating* depositions introspectively, or reflectively, i.e., the annotation could have been done by someone who did not do the problem formulation, or by someone who did it himself. A couple of observations can be listed from (removed for blind review):

- The novice group described a trade-off between weight and speed of the plane and related thrust and lift functions to their corresponding behaviours through the Bernoulli physical rule ("… with the weights that we have and the power system [selected] we'll be able to determine the velocity required to get the lift needed …".
- Knowledge about key issues was evident when the expert explicitly drew relations among many fragments early on. He quickly pointed out that "… the ratio of the wing surface to plane speed should be in this area …", referring to a 'load-speed' chart in aircraft design. Such insight was absent among the novices.

2.2 Shortcomings of initial Problem Maps

Even though our first Problem Maps helped us highlight differences among different designers for two different problems (removed for blind review), there was redundancy and a lack of a uniform structure in the ontology. The *Trade-off* entity, for example, could have been expressed in terms of relations among other entity types, here *Parameters*. A hierarchical structure was also missing. The entity *Structure* entity was created for the purpose of capturing data fragments about a tentative product architecture. However, hierarchies should not have been limited to solution principles or components. Creating function structures is a well-established method in early conceptual design.

3 THE UPDATED PROBLEM MAPS ONTOLOGY

After the first ontology which lacked the hierarchical structure and inter-entity relations that are common building blocks of an ontology, two steps were taken to revise the Problem Map ontological framework. First, an exhaustive list was created where entities relevant to early conceptual design were enumerated either from introspection, or studying the literature. These entities where then organized in multiple iterations to group semantically close entities while shrinking the size of the list, see Figure 2.

The second step was to come up with a final list, and create a common structure that resembled an ontology. A summary of the result can be seen in Table 1. The updated Problem Map ontology consists of six groups of entities, their sub-types, optional attributes that describe some of the entities, a hierarchical structure (in the form of parent-child relations), and inter-group relations.

Table 1. Elements of the updated Problem Map Ontology

Entities	Requirement	Use scenario	Function	Artefact	Behaviour	Issue
Sub-types	Objective Specification	User Environment	-	Solution principle Physical embodiment	Equation Parameter	Conflict Question
Attributes	Objective weight Spec range	-	Parameter	Parameter	-	Importance

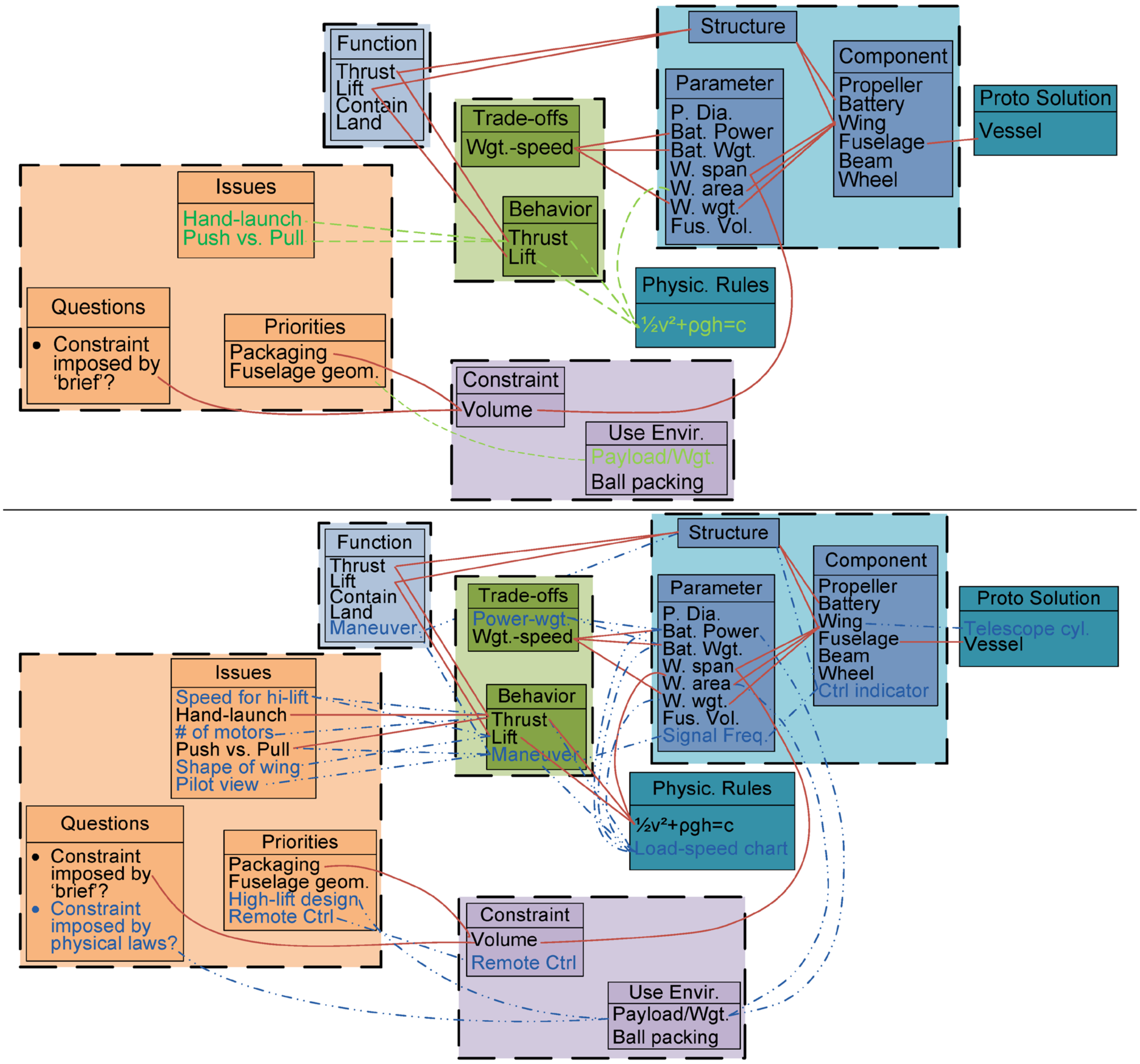

Figure 1. Comparisons of Problem Maps for the design of a model aircraft: top-novice, bottom-expert from (removed for blind review)

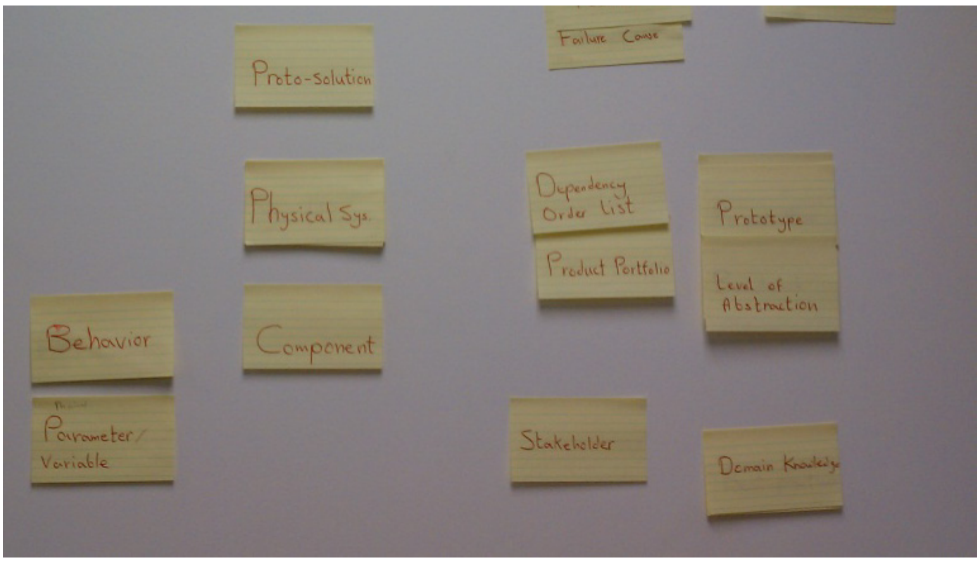

Figure 2. Grouping semantically close entities from an exhaustive list of relevant entities

Requirements define what the design should achieve and what its major goals are; this can come from the initial problem statement, or be implicit requirements that designers identify. Hierarchies of requirements can form requirements or objective trees with assigned weights. *Use scenarios* explain the use environment and target users where the design will be used. *Functions* refer to what the design does in terms of action verbs, e.g., amplify torque. Function hierarchies can be taken as functional decompositions. The P-maps model incorporates disjunctive composition, making it possible to have multiple functional decompositions using common sub-functions. *Artefacts* are the objects that describe the physical components of the design or the concepts the design may be using. *Behaviours* describe the physical properties and laws that govern the design. They can be equations and the parameters that are relevant to both artefacts and functions. Finally, *issues* are entities that describe the problems associated with other entities in the formulation. Examples are conflicting requirements, the improbability of an artefact achieving a function within the bounded requirements (constraints), or questions about missing information. An example of an updated Problem Map can be seen in Figure 3; the data fragments express a design problem about building an autonomous surveillance device that can move around two building in a figure ∞ starting from the centre between them and finishing as closely as possible to the starting point. The data fragments are entered by the designers in a web-based data collection testbed based on the ontology.

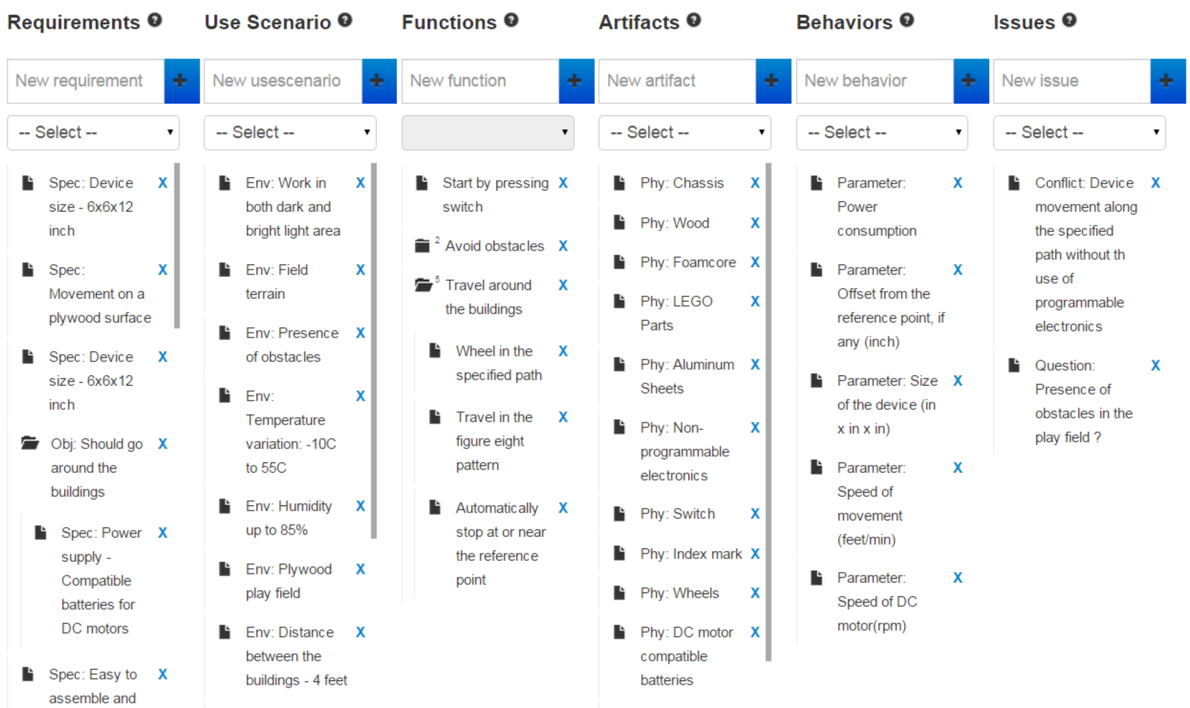

Figure 3. A snapshot of an example of updated Problem Map

4 DISCUSSION

4.1 Clarifying the definitions of entities

This section describes some of the challenges that we have faced in developing and implementing an ontology for expressing problem formulation. Three major sources of confusion have been identified. They make it difficult to rely on data collected from students without being validated by an expert judge or coder:

1. Close definitions of some entities or their generality – One of the major causes of confusion among both introspective and reflective annotations is the general nature of some of the entities. Requirements for example can be a description of anything that might fall under the other categories; a requirement can be about which age group uses the final product (use scenario), or constraints on acceptable materials (artefacts).
2. Similarity of relations between two types of entities and attributes of some entities – Another point of confusion arises from the fact that the description of some entities can be easily

artefact.

mistaken with their relations to other entities. For example, sometimes it is difficult to make a distinction between an attribute of an artefact that is actually a description of the form or the size of the artefact with parameters that in our model fall under behaviours and relate to the said artefact.

3. Counterfactuals – Expressions of negations, such as not using a type of artefact, are common especially in early stages of conceptual design where problem definition continuously changes and some fragments are omitted or modified.

4.2 Striking the balance between structure and fluidity

Striking the right balance between a structured representation and a fluid and free form expression of thoughts has always been a challenge in our development of the ontology. The structure influences both the data collection and the search for patterns in annotated data. The more elements in the ontology, the more difficult it will be to learn by both introspective and reflective annotators. In addition, more elements lead to more combinations in patterns to search which causes lower chances of discovering patterns (the curse of dimensionality). Inspired by Concept Maps (Novak and Gowin, 1984), we initially drew our data fragments as a Concept map, see Figure 4. There were two major problems that can easily be seen in this concept map: there is no distinction between actions as process data, and state data; the representation is highly unstructured and there is a lesser chance of understanding and highlighting differences among designers.

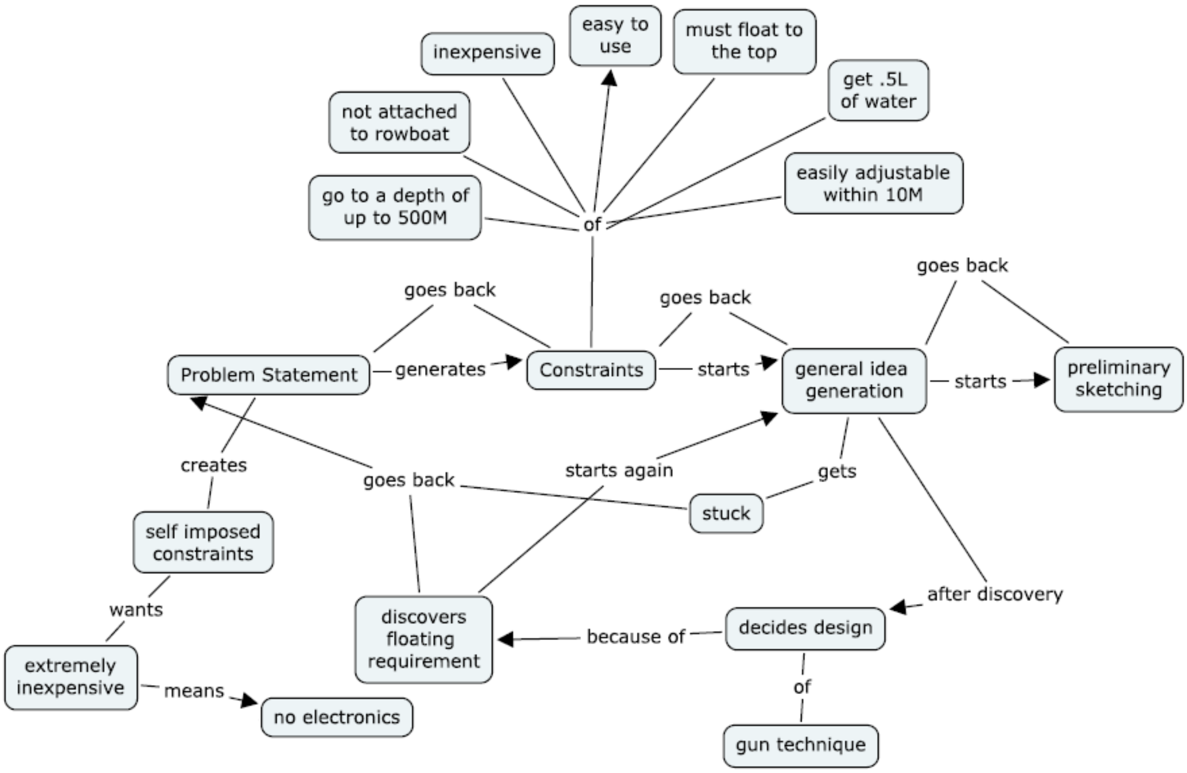

Figure 4. An initial Concept Map drawn to represent problem formulation data for a water sampling device

4.3 Making the learning of the ontology intuitive to users

As we established above, the richer the ontology becomes, the more difficult it will be learned by introspective and reflective annotators. For the former, we can provide evidence of lower agreement among annotators with a measure of inter-rater agreement, Fleiss's Kappa (Fleiss, 1971). Three annotators were given 48 segments produced by other designers. When they were asked to choose one of the major entity types for each segment, the agreement was 0.48 which is a moderate agreement. When the annotators were asked to specify sub-types and relations, the agreement dropped to 0.35. The results were the same for more familiar annotators (researchers who collaborated in developing the initial ontology); agreement fell from 0.75 to 0.64.

Some difficulties for reflective annotators was found from a group of students who designed a prototype for the figure ∞ (also called "figure eight") problem (Figure 3). Reflective interviews showed that some students had not only put fragments into wrong entities, but they had used vague terms which could be fit in other categories. For example, most students misunderstood or misused behaviour entity among six different entities. Even though the students were instructed to consider that what the vehicle does is not behaviour but function, they used "behaviour" to capture the motion of the vehicle: "move in figure eight"; "turn accurately around two blocks"; "circle as quickly as possible".

On the other hand, a word which has too broad a meaning might be used in multiple entities. For instance, many students put the word "autonomous" under *requirement*, *behaviour*, and *issue*. Generally speaking, "autonomous" can be fit into those entities within the ontology. However, physical principle and parameter which describe the detail of an autonomous vehicle should go into behaviours.

5 REVIEW OF OTHER DESIGN ONTOLOGIES

Since Gero (Gero, 1990) initially identified Function (F), behaviour (B), and structure (S) modelling of design objects and design thinking process, other researchers have developed similar variations with specific purpose. Therefore, this section reviews previous various representation frameworks: Function (F)-behaviour (B)-structure (S) (Gero, 1990) and situated F-B-S of Gero and Kannengiesser (Gero and Kannengiesser, 2004); Functional Representation (FR) of Chandrasekaran (Chandrasekaran, 1994); Structure (S) -Behaviour (B) - Function (F) of Goel et el. (Goel et al., 2009); Function (F)-Behaviour (B) State models of Umeda et al. (Umeda et al., 1990, 1996); causal behavioural process of Srinivasan and Chakrabarti (Srinivasan and Chakrabarti, 2009).

F-B-S ontology framework of Gero (Gero, 1990) is a knowledge representation schema for design object. F-B-S model provided explicit representation model of functions to organize knowledge of structural behaviours. Since then, there is a general consensus that F-B-S framework is appropriated in modelling the design process (Gero, 1990), as a coding schema in protocol analysis (Pourmohamadi and Gero, 2011; Suwa et al., 1998), and for design automation (Anthony et al., 2001). Despite the fact that the F-B-S framework predominantly used in design research, it was limited a high-level model. Gero and Kannengiesser (Gero and Kannengiesser, 2004) hence introduced the situated F-B-S framework, which is an extension of the FBS framework, so as to explicitly capture situated cognition in designing.

Similar to F-B-S framework, Chandrasekaran (Chandrasekaran, 1994) introduced Functional Representation (FR) that employed functional representation schema in design structure. That is, FR model not only clarified the meanings of device using function and structure, but this device ontology also provided explicit representation of device sufficiently. Inspired by Chandrasekaran's (1994) Functional Representation (FR) scheme, Goel et el. (Goel et al., 2009) also developed the Structure-Behaviour- Function (S-B-F) modelling language for a teleological description of complex systems. S-B-F explicitly modeled a device's structure with behaviour and function. Behaviour plays a role as a causal link between structure and function. More recently, Helms and Goel (2013) have defined a grounded knowledge representation for the documentation and search of biologically inspired solutions to design problems, and proposed the Four-box method to improve search among sets of conditions of problem formulation, and to evaluate retrieved analogies in biologically inspired design (Helms and Goel, 2014).

Umeda et al. (Umeda et al., 1990, 1996) developed Function Behaviour-State models, whereby subjective function is clearly differentiated from objective function. Then this function is defined as description of behaviour. Finally, state is defined as the entities, their attributes, and the relationships among them. As a result, Function, Behaviour, relationships from FB State models provides a hierarchical structure.

There is not a clear definition of what on ontology is in a design research, since historically it has been a concept in philosophy. A conventional definition of an ontology is a taxonomy plus inter-category relations, i.e., a taxonomic structure that represents knowledge with defined relation types among the categories of the taxonomy. Uschold (1998) defines an ontology in the following: "An ONTOLOGY may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are interrelated which collectively impose a structure on the DOMAIN and constrain the possible interpretations of terms."

In engineering design research, different ontologies have been proposed with either generic or specific scopes of applications. Sim and Duffy (2003) have defined a generic ontology of engineering design activities by creating a structure for a set of steps in a general design process, and for design generation, evaluation, and management activities. Each step includes four elements which may be related in a specific way: the goal of the design activity (G_d), the input knowledge (I_k), the output knowledge (O_k), and the knowledge change. For example, for the design activity of abstracting, the four mentioned are as the following respectively: to simplify the complexity of the design object (Gd); types of abstraction (I_k); appropriate abstractions of design object, e.g., sketches (O_k); and knowledge abstractions that depict useful relationships of the evolving design concept. The objective of such ontology is creating a coherent interpretation of definitions of the activities in order to have more effective design support.

Srinivasan and Chakrabarti (2009) developed SAPPhIRE to explain the knowledge of biology and artificial system problems with a generic causal behavioural model. As a knowledge representation model, this model can be termed SAPPhIRE (State(S), Action(A), Part(P), physical Phenomena (Ph), Input(I), oRgan (R), and Effect (E)). Relationship among SAPPhIRE constructs provides linking function, structure, and behaviour with a generic causal behavioural model. Based on this representation (Srinivasan et al., 2013), they developed an ontology by building clusters of nouns, verbs, adjectives, adverbs and mathematical equations from their earlier work with the SAPPhIRE model. They identified relationships between the clusters and the constructs of the model, and validated the ontology by comparing it to other ontologies.

One approach in developing ontologies is to use standard modelling languages. Wölkl and Shea (2009) have used SysML in modelling conceptual design. They follow the prescribed systematic engineering approach by Pahl and Beitz (1996). They use SysML elements to define entities in conceptual design: Requirement diagram correspond to new specifications, Use Case diagram and Activity diagram describe functions, and Block diagram is used for working principles. Using such a standard language makes it easier to integrate the often non-geometrical data of conceptual design with later stages of product development. However, Wölkl and Shea (2009) state that the representation is not easy to use since multiple (and separate) diagrams are required to represent different aspects of the designs.

Other relevant applications of ontologies include developing an ontology of generic engineering design activities (Sim and Duffy, 2003), and a thesaurus that captures an Engineering-to-Biology process in engineering design (Nagel et al., 2010). We should also mention a similar work to our Concept-map-like ontology; Think maps (Oxman, 2004), where the medium is a non-hierarchical concept map (no structure similar to our ontological framework), and the objective for the ontology is teaching domain knowledge (comparison of a student's map to that of a teacher or norm). The similarities to our ontology are using a computational framework (method), and educating students by comparing them to a normative knowledge structure (application).

As discussed above, various schemes have been applied to representing specific knowledge domain. Nevertheless some common characteristics can be summarized: many studies mainly were concerned about how to explicitly represent design thinking with function, behaviour, and structure. Other studies created new ontology frameworks or hierarchy systems as needed. These align with our research goal of developing the P-maps ontology for studying problem formulation.

6 CONCLUSIONS

The need for capturing and documenting problem formulation data exists. We reviewed two versions of our ontology for expressing problem formulation data. We also reviewed existing ontologies or frameworks, and pointed out that a majority of them are at the extremes of the line between highly structured-highly fluid representation. We believe that our ontology strikes the right balance to highlight difference among different types of designers. However, it faces some challenges in having clear boundaries of definitions for entities, and making it easier for users of the ontology (introspective and reflective annotators) to learn it. The design thinking community can benefit from an effort in reconciling different ontologies for expressing early conceptual design.

REFERENCES

Anthony, L., Regli, W.C., John, J.E. and Lombeyda, S. V. (2001), "An Approach to Capturing Structure, Behavior, and Function of Artifacts in Computer-Aided Design", *Journal of Computing and Information Science in Engineering*, Vol. 1 No. 2, pp. 186–192.

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*, Vol. 40 No. 7, pp. 801–811.

Chandrasekaran, B. (1994), "Functional Representation: A Brief Historical Perspective", *Applied Artificial Intelligence*, Taylor & Francis, Vol. 8 No. 2, pp. 173–197.

Danielescu, A., Dinar, M., Maclellan, C.J., Shah, J.J. and Langley, P. (2012), "The Structure of Creative Design: What Problem Maps Can Tell Us about Problem Formulation and Creative Designers", *Proceedings of ASME IDETC/CIE*, Chicago, IL, USA.

Dinar, M., Shah, J.J., Langley, P., Hunt, G.R. and Campana, E. (2011), "A Structure for Representing Problem Formulation in Design", in Culley, S.J., Hicks, B.J., McAloone, T.C., Howard, T.J. and Chen, W. (Eds.),*Proceedings of the International Conference on Engineering Design*, Copenhagen, Denmark, pp. 392–401.

Dorst, K. and Cross, N. (2001), "Creativity in the design process: co-evolution of problem–solution", *Design Studies*, Vol. 22 No. 5, pp. 425–437.

Doumont, J.-L. (2002), "Verbal versus visual: A word is worth a thousand pictures, too", *Technical communication*, Vol. 49 No. 2, pp. 219–224.

Fleiss, J.L. (1971), "Measuring nominal scale agreement among many raters.", *Psychological bulletin*, American Psychological Association, Vol. 76 No. 5, p. 378.

Gero, J.S. (1990), "Design prototypes: a knowledge representation schema for design", *AI Magazine*, AAAI, Vol. 11 No. 4, pp. 26–36.

Gero, J.S. and Kannengiesser, U. (2004), "The situated function-behaviour-structure framework", *Design Studies*, Vol. 25 No. 4, pp. 373–391.

Goel, A.K., Rugaber, S. and Vattam, S. (2009), "Structure , Behavior and Function of Complex Systems: The Structure, Behavior, and Function Modeling Language", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 23 No. 1, pp. 23–35.

Helms, M. and Goel, A.K. (2013), "Grounded Knowledge Representations for Biologically Inspired Design", *Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 6: Design Information and Knowledge*, Design Society, Seoul, South Korea, pp. 351–360.

Helms, M. and Goel, A.K. (2014), "The Four-Box Method: Problem Formulation and Analogy Evaluation in Biologically Inspired Design", *Journal of Mechanical Design*, Vol. 136 No. 11, p. 111106.

Larkin, J.H. and Simon, H.A. (1987), "Why a Diagram is (Sometimes) Worth Ten Thousand Words", *Cognitive Science*, Blackwell Publishing Ltd, Vol. 11 No. 1, pp. 65–100.

Maher, M. Lou, Poon, J. and Boulanger, S. (1996), "Formalising Design Exploration as Co-Evolution: A Combined Gene Approach", in Gero, J.S. and Sudweeks, F. (Eds.),*Advances in Formal Design Methods for CAD: Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for Computer-Aided Design, June 1995*, Springer US, pp. 3–30.

Maher, M. Lou and Tang, H. (2003), "Co-evolution as a computational and cognitive model of design", *Research in Engineering Design*, Vol. 14 No. 1, pp. 47–63.

Miller, G.A. (1995), "WordNet: a lexical database for English", *Communications of the ACM*, Vol. 38 No. 11, pp. 39–41.

Nagel, J.K.S., Stone, R.B. and McAdams, D.A. (2010), "An engineering-to-biology thesaurus for engineering design", *Proc. 2010 ASME IDETC/CIE, Montreal, Quebec, Canada*.

Novak, J.D. and Gowin, D.B. (1984), *Learning how to learn*, Cambridge Univ Pr.

Oxman, R. (2004), "Think-maps: teaching design thinking in design education", *Design Studies*, Vol. 25 No. 1, pp. 63–91.

Pahl, G. and Beitz, W. (1996), *Engineering Design: A Systematic Approach*, Springer, London, UK.

Pourmohamadi, M. and Gero, J.S. (2011), "LINKOgrapher: An Analysis Tool to Study Design Protocols Based on FBS Coding", *Proceedings of the International Conference on Engineering Design*, Copenhagen, Denmark, pp. 1–10.

Sim, S.K. and Duffy, A.H.B. (2003), "Towards an ontology of generic engineering design activities", *Research in Engineering Design*, Vol. 14 No. 4, pp. 200–223.

Smulders, F.E., Dorst, K. and Reymen, I.M. (2009), "Modelling co-evolution in design practice", in Norell Bergendahl, M., Grimheden, M., Leifer, L.J., Skogstad, P. and Lindemann, U. (Eds.),*Proceedings of the International Conference on Engineering Design*, Palo Alto, CA, USA, pp. 335–346.

Srinivasan, V. and Chakrabarti, A. (2009), "SAPPHIRE – AN APPROACH TO ANALYSIS AND SYNTHESIS", *Proceedings of the International Conference on Engineering Design*, pp. 417–428.

Srinivasan, V., Chakrabarti, A. and Lindemann, U. (2013), "Towards an Ontology of Engineering Design Using SAPPhIRE Model", in Chakrabarti, A. (Ed.),*CIRP Design 2012 SE - 3*, Springer London, pp. 17–26.

Suwa, M., Purcell, T. and Gero, J.S. (1998), “Macroscopic analysis of design processes based on a scheme for coding designers’ cognitive actions”, *Design Studies*, Vol. 19 No. 4, pp. 455–483.
Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. and Tomiyama, T. (1996), “Supporting conceptual design based on the function-behavior-state modeler”, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 10 No. 4, pp. 275–288.
Umeda, Y., Takeda, H., Tomiyama, T. and Yoshikawa, H. (1990), “Function, behaviour, and structure”, *Applications of artificial intelligence in engineering V*, Computational Mechanics Publications and Springer-Verlag, Berlin, Vol. 1, pp. 177–194.
Uschold, M. (1998), “Knowledge level modelling: concepts and terminology”, *Knowledge engineering review*, Cambridge University Press, Vol. 13 No. 1, pp. 5–29.
Willows, D.M. (1978), “A picture is not always worth a thousand words: Pictures as distractors in reading.”, *Journal of Educational Psychology*, American Psychological Association, U Waterloo, Canada, Vol. 70 No. 2, pp. 255–262.
Wölkl, S. and Shea, K. (2009), “A computational product model for conceptual design using SysML”, *Proceedings of ASME IDETC/CIE*, San Diego, CA, USA.

ACKNOWLEDGMENTS

This study is supported by the United States National Science Foundation, grant number 1002910. The opinions expressed in this paper are those of the authors and are not endorsed by the National Science Foundation.