Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2011
August 29-31, 2011, Washington, DC, USA

DETC2011-48396

TOWARDS A FORMAL REPRESENTATION MODEL OF PROBLEM FORMULATION IN DESIGN

Mahmoud Dinar
Design Automation Laboratory
Department of Mechanical and Aerospace Engineering
Arizona State University
Tempe, AZ 85287-6106 USA

Jami Shah
Professor
Department of Mechanical and Aerospace Engineering
Arizona State University
Tempe, AZ 85287-6106 USA

Glen Hunt
Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809 USA

Ellen Campana
Assistant Professor
School of Arts, Media, and Engineering
Arizona State University
Tempe, AZ 85287-8709 USA

Pat Langley
Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809 USA

ABSTRACT

Studies on design, show that problem formulation plays a major role in creative design. We plan to construct an interactive computer system that aids problem formulation. In the current stage, to improve our understanding of problem formulation, we have conducted exploratory protocol studies of novice designers and collected data from an expert designer in the form of a depositional interview. A formal representation of the design problem is needed to improve our empirical investigation. We propose a preliminary framework for such a model and we call it a problem map. It provides a basis for comparing how different designers perceive a problem. Our study is based on the design of a model aircraft for the AIAA student design competition. This preliminary analysis shows the evolution of the problem and the solution spaces in the elaboration of the problem maps through time. The problem maps also show a richer representation of attended attributes and relations for the expert and more attributes left in vacuum for the novices.

INTRODUCTION

The focus of research in design theory has been mostly set on the design process. It should be shifted to other aspects: the designer, context and the design itself [1]. Simon regarded the work of the designer as creating an interface for the design and the context [2]. This suggests that the designer should be the center of attention and more specifically, how he or she formulates the problem. Harfield calls this ’Problemization’ and attests it has not received much attention in the literature [3]. Review of the literature implies that a few studies have focused on representing the problem and the solution space, and some on the process of problem formulation. We attempt to bridge the gap in understanding problem formulation and find a way to represent the co-evolution of problem and solution. In addition, a formal model can become a research tool to aid empirical studies of problem formulation in design. We will show that our proposed structure captures the designer’s understanding of the problem at an instance of time, represents its evolution in the elaboration of the model through time, and gives a basis for comparing it to other designers’ understanding of the same problem.

A few researchers have attempted to develop models for representing the structure of design problems. Maher et al [4] link problem definition states to solutions. Goldschmidt draws linkographs of figural representations as states and conceptual representations as operators [5]. Eisentraut and Gunther code sketches for concreteness and completeness as a measure of level of abstraction [6], Fricke implements temporal models [7] and Goel and Pirolli choose a Task-Operator-Phase model [8]. Akin and Chengtah suggest that modes of representation can be extended from verbal and visual to more activity modes such as examining, drawing or writing, thinking, and their echo, speaking. They observe that data modalities are dependent, and novel decisions occur in multi-modal episodes [9].

Research has shown a few characteristics of problem formulation, mainly through protocol studies. Designers prefer to treat problems as ill-defined [10,11]. Atman et al [12] state that senior designers produce higher quality designs by gathering more information early, consider more alternative solutions and transition more frequently between design steps while Eisentraut maintains that such behavior relates to different styles of problem solving, which are independent of the situation of the design episode [13].

It is well known that unlike well-defined problems, design problems continue to evolve throughout the problem solving process. It is suggested that recognition of partial structures in the problem space, shape the structure of the solution space [4, 14]. Cross and Cross [11] claim that creative designers, holding experience of previous solutions at the back of their minds, use first principles as stimuli to build bridges between problem and solution space through key concepts, while Harfield claims that designers need ’proto-solutions’ to compare the goal and the problem state [3]. He also adds that naive designers make fixed assumptions while creative designers question requirements. Others suggest designers set boundaries for particular aspects of attention in framing a problematic design situation [15] and a more successful design team considers more of such framings than an unsuccessful one [16].

In addition, strategies that are adopted in problem formulation have also been studied. Gero and Kannengiesser state that designers use their own experiences to interpret representations augmented with implicit requirements which lead them to the same problem differently [17]. Kruger and Cross categorize designers into problem-driven and solution-driven; or their variants (with higher iterations among stages) information-driven and knowledge-driven [18]. Ho states that expert designers decompose a problem explicitly, approach directly the goal state first, work backward for required knowledge and then forward for solution while novices decompose a problem implicitly and eliminate a problem when they fail to handle it [19]. Ball et al have found that experts lean on schema-driven analogies (experiential abstract knowledge), while novices rely on case-driven analogies – which maps the source problem and solution to a target problem [20]. Gero and McNeill classify the different strategies designers adopt into analysis, proposition and making explicit references (micro strategies) and top-down, bottom-up, decomposition, opportunistic and backtracking (macro strategies) [21].

Our own framework draws on conceptual structures that are common in design and design education, with an emphasis on the details of what the designers’ representation of the problem itself is at any time. We developed the framework based on an empirical study that has been described in prior work [22].

In this paper we describe the framework and apply it to data from a protocol study with a problem that was substantially different from the one that the earlier work was based on. We present examples and details from two participants, and show how the framework is useful for understanding how problem representations differ across participants. More specifically, the expert designer introduces a level of abstraction that leads to a richer representation. We also show how the framework is useful for understanding how the activity of exploring a proto-solution can lead to insights in other aspects of the design. Finally, we discuss how the framework may need to be modified in future work, based on our experience with applying it to a new data set.

A FRAMEWORK FOR REPRESENTING A DESIGN PROBLEM

In this section we describe our framework for representing problem formulation in design, based on our prior work. The framework is designed to represent the designer’s understanding of the problem at each instance of time. Therefore, we consider it a state representation. We often call this representation a problem map (or P-map). The framework is a work in progress, and each time we apply it to a new set of data we gain insight that will improve it for future iterations.

As in our prior work, we may interchangeably use the terms ‘element’, ‘entity’, and ‘class’, referring to a common data object such as *Function* or *Issue*. We call each data record of an entity, an attribute. For example, valve is an attribute of the *Component* entity. We may also refer to a complete or a part of a design session, as a design episode.

In contrast to models that represent design process, a central criterion for our framework is the adoption of a state view. Therefore, we have deliberately avoided elements that correspond to operators, for example “decisions”, in favor of concepts that are directly linked to participants’ understanding of the problem itself at a particular point in time.

In our earlier framework, we grouped entities into the following categories: function, structure, parameter, behavior, issue, usage and knowledge, see Figure 1. The grouping was mostly based on associations found in the literature [21,23,28]. In our findings, we see that such associations do not appear in a prescriptive manner. For example, not all parameters of a component are related to a function through a behavior. Thus we have abandoned the grouping.

There are slight changes in the new *P-map* framework partly in accordance with new observations and partly for simplification. We have removed the entities *System architecture*; we show the hierarchy with relations among components. We have also removed *Tradeoff* and *Priority*. The former can come up as an issue. The latter can be monitored in changes through time. We have added an entity for case analogies (*Analogy*), and have substituted *Constraint* with a more general class of *Requirement*.

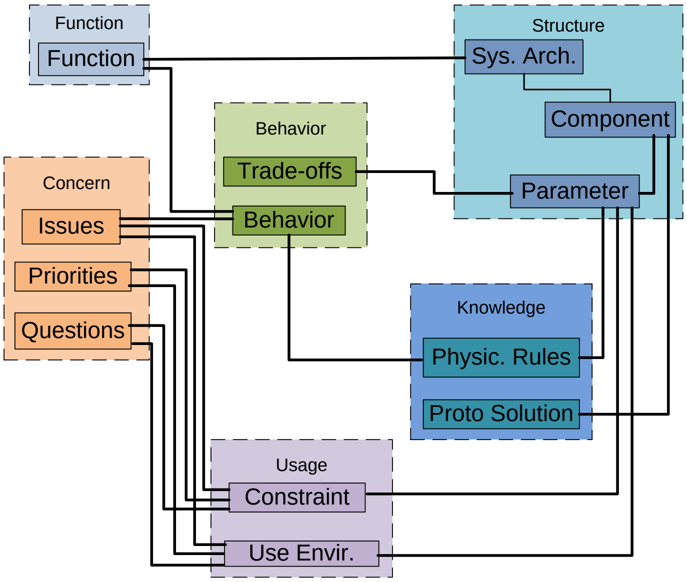

Figure 1 A GENERAL FRAMEWORK FOR THE PROBLEM MAP [22]

EMPIRICAL STUDY

The main objective of our research is to study the influence of problem formulation in design creativity and to build a computer system to aid it. A few of our research questions are: Do creative designers have more abstract representations of a problem than less creative designers, leading to more fruitful designs? If so, how do these abstract representations evolve over time? How might a software system support designers in achieving the right kind of abstraction? How is expertise related to problem formulation? Does it interact with creativity?

To study these issues objectively, we need a structure for organizing the content of the formulation so that we can examine how it evolves at a finer level of granularity than attempted before. We have developed such a framework based on excerpts from two student groups designing a radio-controlled airplane for the AIAA[1] Design/Build/Fly competition, and a depositional interview with an expert in designing RC aircraft [22]. In this paper we apply the framework to "think-aloud" protocol data from participants working on a much simpler mechanical engineering design problem (Figure 2). This is a good test of the generalizability of the framework, because both the problem itself and the type of data differ from the data that the framework was developed on.

The design case

We conducted a protocol study with eight engineering design experts from industry and nine senior undergraduate mechanical and aerospace engineering students. Analysis is continuing. Here we present data from two participants: one expert, one novice.

[1] The American Institute of Aeronautics and Astronautics

Each participant worked on a single problem alone for a maximum of one hour. While working they were asked to verbalize their thoughts without considering whether what they were saying would make sense to someone else. They were allowed to write and/or sketch as desired. Throughout the session an experimenter was present in the room but out of sight. The experimenter's role was to ensure that the session was recorded and to prompt the participant if he or she fell silent. Audio and video of the session was collected and later transcribed.

All of the participants worked on the water sampling problem (Figure 2).

Design a mechanical device to be used from a rowboat by a researcher who wishes to collect samples of water from fresh-water lakes (eg., Lake Tahoe) at known depths down to a maximum of 500 m.

After release, the device must not be attached to the boat and must descent to within 10 m of an easily adjustable pre-determined depth. It must return to the surface with a 0.5 liter sample of water from that depth and then float on the surface until picked up.

The device should be reliable, easy to use, reusable, and inexpensive.

Figure 2 THE DESIGN PROBLEM THAT PARTICIPANTS WORKED ON WHILE "THINKING ALOUD" IN OUR PROTOCOL STUDY.

Observations

We applied our framework to the protocol data by going through the transcripts and notating fragments that the participant mentioned with the corresponding entities, and instances where the participant implied connections between these entities. In this section we will provide examples of this.

We explain in detail the fragments of the design episode for the expert designer. We present the emergence of new attributes, dependence of attributes within an entity and among entities (covariation), emergence of new ideas for similar concepts (option), instantiation of attributes based on previous observations (instantiation), and substitution of previous attributes (substitution).

In summary, the expert designer starts with considering abstract concepts (proto-solutions) such as tethering. Then he sketches some components based on the proto-solutions, relates some of them through behaviors to realize functions e.g. relating the pressure behavior of a cylindrical component to the ascending and descending function of the water sampler. He encounters issues either when he examines the abstract proto-solutions or after a more elaborated analysis of the behavior of a component and the underlying physical rules. Table 1 shows a few examples of the attributes corresponding to the elements of our framework. The first column shows the order of occurrence of each utterance. The second column is the corresponding entity. The third column is the extracted quote from the protocol. The next column is the label with which we show the attribute in the *P-map*, and finally the last column shows the related observations.

We dissected the protocol based on the pauses the designer took. The total number of utterances for the whole design episode was 108. The selected observations in Table 1 are taken from the first 13 minutes of a session which lasted 52 minutes.

Table 1 EXAMPLES FOR THE OBSERVED ELEMENTS AND THEIR RELATIONS

No	Entity	Observation	Label	Relat.
2	Function	*it just needs to ascend and descend*	F: Descending F: Ascending	
4	Proto-sol.	*[concept] B is some sort of depth indicator*	PS: Depth indicator	
7	Proto-sol.	*So [concept] B is some sort of depth transducer*	PS: Depth transducer	4
12	Proto-sol.	*then the other main subsystem is the sampling chamber*	PS: Sampling chamber	
13	Component	*there is a hollow cylinder and one end is capped and there is a piston*	C: Cylinder	12
14	Behavior Parameter	*and this piston since this is filled with atmospheric air on the backside of the piston, atmosphere, atmospheric pressure you will pre-determine how far this piston travels which will determine the depth*	B: Cylinder-pressure P: Piston-displace.	2
15	Physical rule	*we know that about 34 feet of freshwater is one atmosphere so you can determine how many atmospheres of compression that you want the system to move before you trigger*	Ph: Depth-pressure	14
17	Function	*it tells the sampling to go ahead and take the sampling*	F: Triggering F: Sampling	16
19	Component Function Parameter	*it has a diaphragm that ruptures at a specific pressure*	C: Diaphragm P: Rupture-pressure	15,17, 18
20	Parameter	*that diaphragm ruptures when it gets down to a pre-determined depth*	P: Depth	19
24	Physical rule Usage Issue	*500 meters is about 1500 feet so that is well beyond the limits of normal air and nitrogen mixture they will have nitrogen narcosis*	Ph: Air-mix-depth U: Diving-depth I: Nitrogen-narc.	23
26	Proto-sol.	*So the tethering can obviously be, it can be electronic*		3,8
27	Comp Function	*it could have some sort of encoder that meters out the cable and some sort of motor*	C: Encoder C: Cable C: Motor F: Metering	26
28	Behavior Parameter	*keeps track of the amount of cable that is extended*	B: Track-cable P: Cable-length	27
29	Function Comp Parameter	*And then when the sampling device gets to a certain depth it can have a wire that is wound into the cable that opens the sampling and can have a saw that opens the sampling uh, sampling valve*	C: Sampler C: Wire C: Valve P: Saw-form	17
38	Function Behavior Parameter	*So when the device is going down the last 10 feet of the cable is a steel rod, perhaps and then it transitions, the cable transitions into the rod for some period of distance. And then the sampling device uses the angle between the rod and the sampling device to open a valve*	C: Rod P: Rod-angle	27,29, 37

Representing relations in the framework

We have presented before, a framework for representing a P-map [22]. We used existing relations corresponding to models described in the literature e.g. [23], and assumed relations exist implicitly among all similar instances of the same entities. In this paper, we abandon that prescriptive model. We choose to show only explicit relations. Such representation might be a vehicle to show styles and strategies among designers. For example, we see that a designer relates a parameter directly to a function without examining the behavior that governs realizing that function. Such bypassing can represent a higher level of abstraction in an early stage of a design.

The numbers in the last column of Table 1 refer to the related attributes. For simplicity, we do not show the labels in the previous column when there is a reference to a previous instance in the last column. For example in utterances 19 and 29 the function is referred to utterance 17. Similarly, in utterance 26, a relation exists between previously defined attributes and thus there is no label.

We have identified four major types of relations. Attributes of an entity may depend on the attributes of the same entity or other entities (covariation), for example rupture pressure, depth, and the triggering function are correlated. Emergence of new ideas for similar concepts is considered as addition of an option, for example in utterances 29 and 38, the wire is used in one design with a valve and in another design with a rod.

Another type of relation occurs when an attribute is an example of an earlier attribute. A proto-solution in utterance 12 is immediately instantiated as a component in the next utterance. Finally we have identified a relation type when an attribute substitutes an older attribute. The proto-solution in the fourth utterance is replaced with a yet abstract proto-solution in utterance 7.

We initially looked for other ways to represent P-maps as we will explain later. The relations in our P-map resemble parent-child relations in ERD[2] models (e.g. 'Physical system architecture' and 'Component'), class structure relations in UML[3] models (e.g. 'Physical system architecture' and 'Function') or optional attribute relations in EXPRESS-G [25] models (e.g. 'Use environment' and 'Priority'). However they are not limited to those relations and they cannot be entirely defined in either of those models. For example, in a class structure of a UML model, 'Function' would be a mechanism of a class of a certain object with the 'Physical system architecture' or 'Component' as its attributes. We discuss this later when we talk about representations in other disciplines. In the next section, we will explain the presented model through examples from our empirical data.

[2] Entity Relationship Diagram
[3] Unified Modeling Language

ADVANTAGES OF PROPOSED REPRESENTATION

In this section we give examples of the P-maps of our case study and describe what benefits we get from the proposed representation. We should emphasize that we do not claim too confidently that our representation illustrates the differences between experts and novices, as the examples given are based on the analysis of two protocols. However, as we mentioned earlier, investigating such differences is not a main objective of our research at the moment. Nevertheless our P-maps have no differences in the type of content or structure for either of the subjects. It is also similar to our previous work [22] with a minor changes and this can be viewed as a potential benefit of the P-map which can represent design data, regardless of the method of data collection.

P-maps of the case study

In this section we present the *P-maps* for an expert designer and a novice who had some experience with a similar problem. His familiarity however does not affect the results dramatically as we will see, there are relatively few references to domain knowledge (the entities physical rules and analogy specifically).

In our previous work, for the sake of simplicity, much of the attributes of each entity were omitted and an overall *P-map* at the end state was represented. Here, however, we will focus on the elaboration of *P-maps* and show a complete representation in a shorter period of time, about a quarter of the design session. We try to use the same names for both designers whenever there is no semantic difference to give a better opportunity for comparison.

Figure 3 shows the *P-maps* for the expert designer. Figure 4 shows the *P-maps* for the novice designer. As it can be seen, there are a few entities that are not present in these *P-maps* for either of the designers (e.g *Requirements* for the expert and *Physical rules* for the novice), or are absent for both designers (e.g. neither of the designers have an example of the entity *Question*). The absence is due to the fact that the instances occurred in later stages of the design episode. For example, the expert designer revisited the requirements after about 19 minutes and raised a question after about 33 minutes. We believe such observations will be a way of using the *P-maps* to show strategies or patterns as characteristics of a specific group of designers, though we should leave such finding until we have a more concrete analysis for more data.

We can speculate some observation for further analysis from utterances in Table 1 as seen on the *P-maps*. We can see that abstract instantiation occur between two proto-solutions (utterance 26). We can represent function sequences as relations (utterance 17). We can accommodate non-numeric parameters (utterance 29).

Now that we have illustrated how different a designer perceives of a problem at different times and how different designers' mental models are of the same problem in our P-map, we will explain the benefits of our model in more detail.

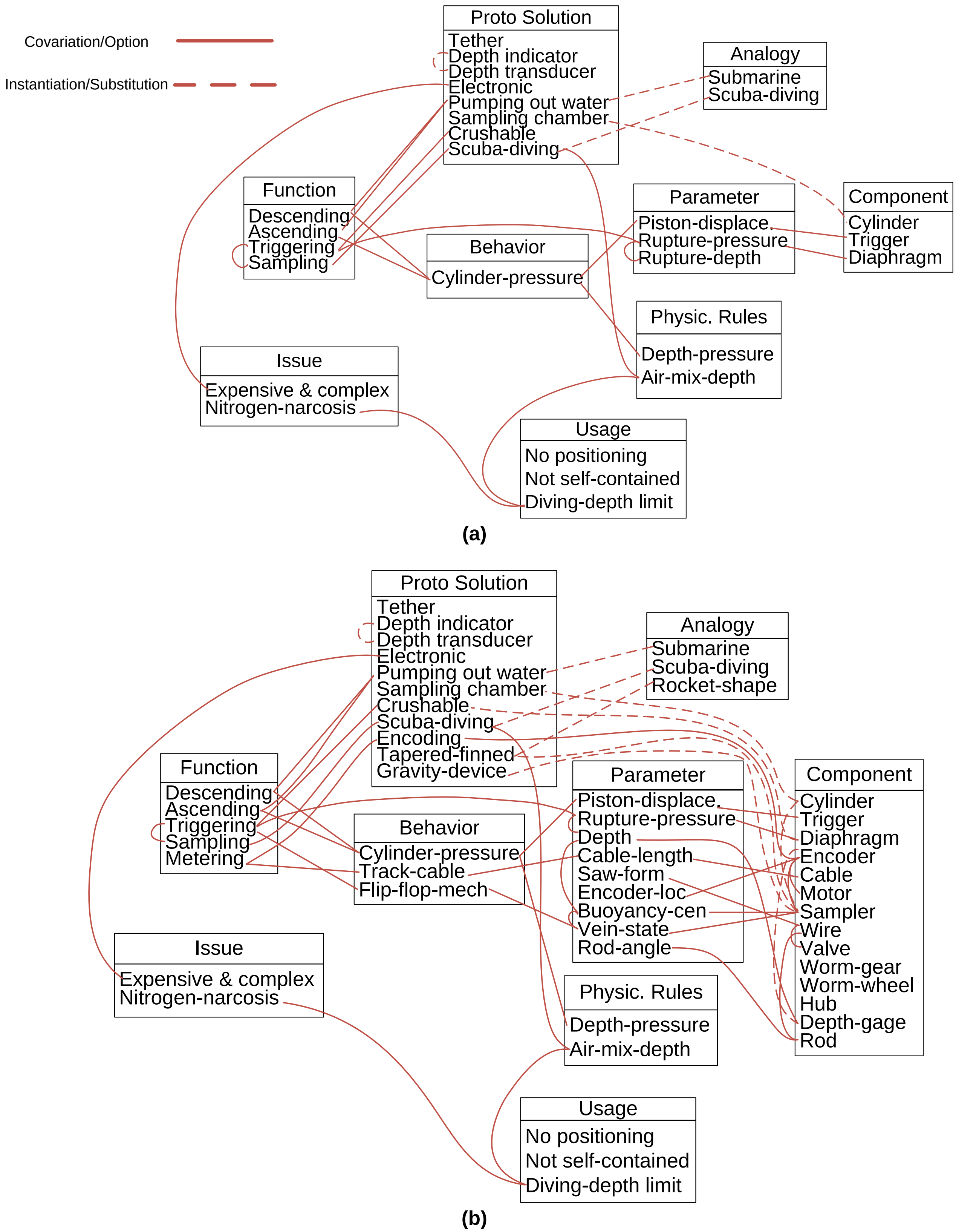

Figure 3. P-MAPS OF THE EXPERT DESIGNER (A) AFTER EIGHT MINUTES; (B) AFTER THIRTEEN MINUTES

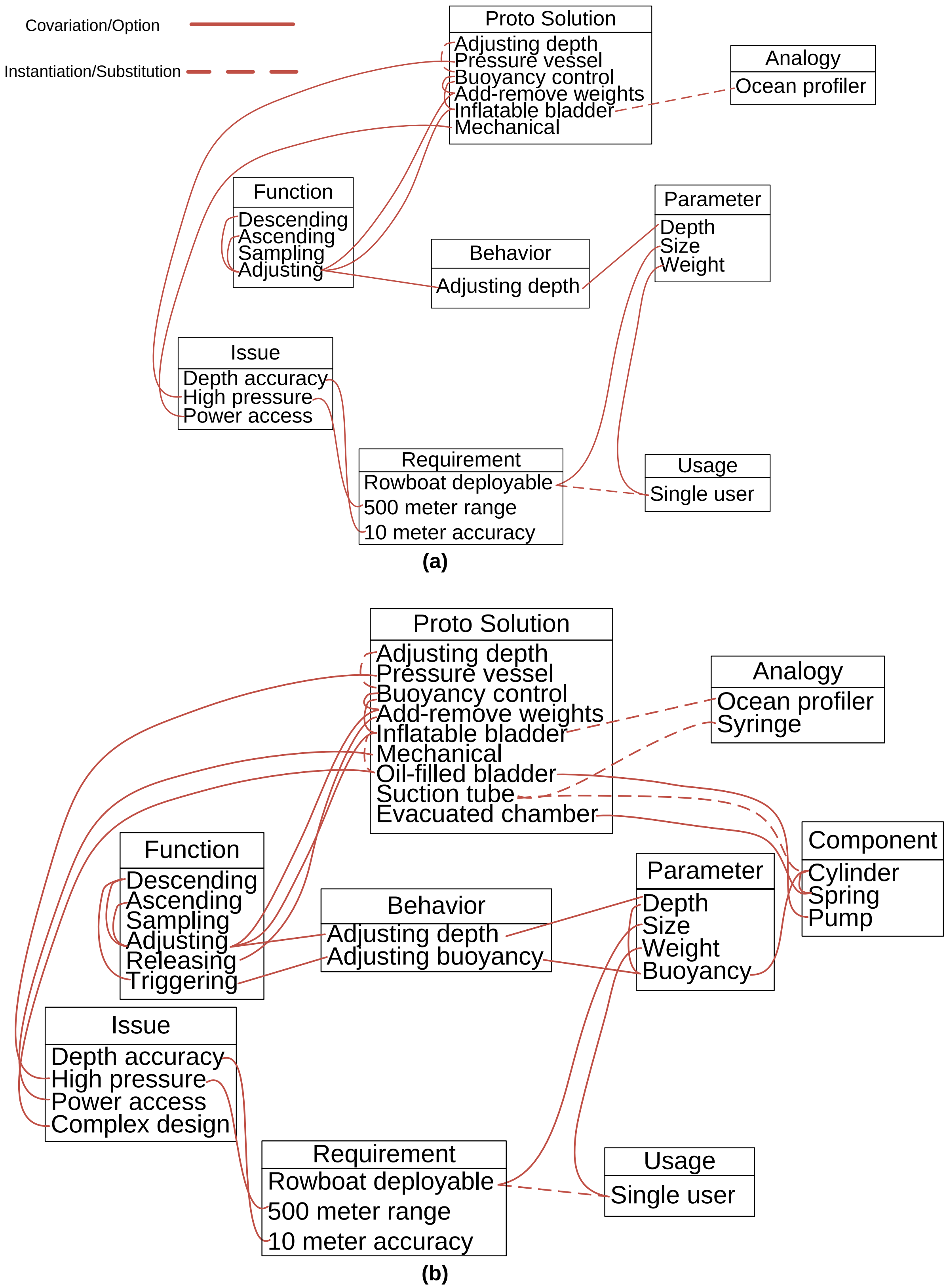

Figure 4 P-MAPS OF THE NOVICE DESIGNER (A) AFTER ELEVEN MINUTES; (B) AFTER SEVENTEEN MINUTES

Benefits of the P-map model

The main benefits of the proposed model of the *P-map* are the ability to show elaboration and emergence of design data during a design episode and the ability to show the differences between the mental models of different designers of the same problem. As it can be easily seen in Figure 3 and Figure 4, the *P-map* can show the progress of the designer's problem formulation through time. P-maps act as the snapshots of such progress. In addition, P-maps can demonstrate the richness of the content of the design data during problem formulation for a creative or experienced designer.

We emphasized on the state representation as one of the objectives of our proposed *P-map*. Such static structure is rich enough for representation since it is complicated by a web of dependencies among relationships. The dynamic structure, however, offers a few simple relations in creating a feasible solution [26]. Main issues, their dependencies and the dominant issues become the starting point in our *P-maps*. One key difference between the expert and the novices, as we mentioned, was the richness of the relations which were captured.

P-maps give a state description of design as opposed to a process description, according to Simon [2]. It can be worth investigating whether the study and the representation of cognitive states would be easier than that of cognitive processes, in which case, P-maps provide a more useful basis for a tentative aid system. It will be easier to show cognitive transition through snapshots of states i.e. P-maps than it is to arduously demonstrate how the transition took place.

Another aspect of the P-map is the way it might help organize the inter-entity and intra-entity relationships. As it was seen in the given examples, one key difference between the expert and the novices was the richness of the relations which were captured. Closer examination of those relations might become a platform to decide, how decomposable design problems are, and possibly lead to a measure of complexity beyond what existing taxonomies offer.

Different levels of granularity might be accommodated with our proposed structure. For example a class of standard units (like length or time) might be amended in search of finer relations in designs, considering dimension-less ratios. From another approach, detecting relations among different classes may have the same result. For example we saw that, proto-solutions are used as behaviors in abstraction, or functions may be directly related to parameters. The main reason for this ability is that the framework resembles an object-oriented data model which is well-established and opens ways to implementation and modification more rapidly.

Finally, the resemblance to the object-oriented data model may help integrate a creative design tool onto a Product Life-cycle Management platform, where with well-known and well-structured design ontology automating design tasks is a matter of technicality [27]. This may be the missing link in offering comprehensive design solution software.

We should note that we have not yet defined solid criteria for validating the effectiveness of our proposed model which we will address in the plans for future work. We think though, the model is fairly compact, supports different levels of abstraction and is rich and flexible enough to represent problem formulation of different complexity level.

OTHER APPROACHES TO FORMALIZATION OF PROBLEM REPRESENTATION

In this section we will mention a few approaches that we considered for problem formulation in design. We will discuss their uses and discuss why they cannot serve our purpose.

We touched on some of the representation models in mechanical engineering design in our introduction. In addition, models have been presented to represent information handling in design. Baya and Leifer [28] created a framework based on information fragments in design activity, level of detail and level of abstraction. Grebici et al [29] investigated a generic question-based approach in triggering improved pathway of design thinking considering a taxonomy of knowledge indexes with entities such as process, function, attribute, product structure, feature, issues and resources.

As we discussed in presenting our framework, we also looked at models in software engineering. UML models might be good at representing a specific class of problems, most probably related to a specific class of artifacts or systems as they excel at avoiding redundancies by utilizing similarities in forms of inheritance. Database models such as Entity Relationship Diagrams are means to cluster data and are more concerned about compact relations in order to respond quickly to queries. For example, the entity 'Trade-off' in our P-map could have been an attribute of a behavior table. Expressiveness is a more important objective than compactness for our representation while in an ERD it is the opposite.

Concept maps were also considered [30]. They however, are much less structured to address our concerns. This becomes a major shortcoming especially when we want to compare different instances of the problem formulation through time, or models of different designers. They also lack the hierarchical structure that we are proposing. The only advantage of the concept map is the ability to accommodate fine levels of granularity.

One of the inspiring works for our research is the area of cognitive psychology. Newell and Simon [31] introduced a problem-behavior graph, which is a graphical depiction of the mental states that a designer goes through (typically on a well-designed problem). They used boxes to represent pieces of working memory, a state the designer had entered, or were considering. Links were used to describe successor relationships. They used their model to analyze their finding from verbal protocols of human subjects solving puzzles and playing games. This is similar to our model, in that it is an explicit graphical representation of the mental state of the designer at different points in time, however, it focuses on a search through a well-defined problem space, and not on the development of problem representation.

Finally, we looked in the artificial intelligence discipline. Starting in 1966, Ross Quillian [32] introduced semantic networks. These were supposed to be a representation of meaning. Initially, they were used to represent the meaning of sentences, with the tentative purpose of natural language processing. Nodes were used for representing concepts, and links for the types of relationships among them. This was a representation of some static situation, so if someone read a sentence, they were supposed to have this representation in their memory. In that sense, it was a graphical representation of the subject's mental state, though it largely consisted of beliefs. Compared to our work, it was more focused on making sense of a situation and less on design or any type of a generative process. Namely, it was context-independent.

As we explained, there have been various models in different disciplines which we investigated, however we have concluded that they cannot be completely implemented to represent problem formulation in design, the main objective of our study. In the next section we conclude our initial findings and present a plan of our future work to address the issues we faced either in our existing model or in the models that we discussed above.

CONCLUSIONS AND FUTURE WORK

In this paper we summarized our framework for representing designers' representations of problems. We showed how this framework was applied to data from a protocol study that was fundamentally different than the data set that was used to develop it. Using detailed examples, we demonstrated that the framework proved useful in at least two ways: 1) it allowed us to compare the novice and expert representations, and to see that in this case the expert introduced a function decomposition that was more productive than the novice's representation and 2) it allowed us to examine how one participants' representation changed over time, specifically how through exploring a proto-solution, new constraints were discovered.

The framework is still very much a work in progress. As we applied it to this data we encountered several issues that we will need to address in future work. One of these relates to peoples' ability to represent, and alternate between, multiple solutions to a problem. Currently we have not incorporated a mechanism for forgetting. We show all the attributes on the same *P-map* unless they are dismissed explicitly. The *P-maps* become very crowded and order and direction of search may be lost. Representing this requires creating relations that span time points. In future work we will explore other options for addressing this problem, for instance including previously explored solutions, but distinguishing between active and dormant representations.

Another example of our existing limitations is that our model represents mostly a reductionist space. It is difficult to expand the search space with addition of a requirement. We have observed that insightful designs usually occur when a statement which could be well regarded as an anti-constraint is added.

The P-map has the potential to become a tool for comparing relative complexity of design problems. In addition, it can demonstrate the contrasting strategies that different designers adopt in problem solving, by investigating the sequence of emerging attributes.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Dorst, K., 2008, “Design research: a revolution-waiting-to-happen”. *Design Studies*, *29*(1), pp. 4-11.

[2] Simon, H.A., 1996, *The Sciences of the Artificial*. MIT Press, Cambridge, MA

[3] Harfield, S., 2007, “On design problemization: Theorizing difference in designed outcomes”. *Design Studies*, *28*(2), pp. 159-173.

[4] Maher, M., Poon, J., and Boulanger, S., 1996, Advances in Formal Design Methods for Cad : Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for Computer-Aided Design, Chapman & Hall, London, UK, Formalising Design Exploration as Co-Evolution: A Combined Gene Approach.

[5] Goldschmidt, G., 1997, “Capturing indeterminism: representation in the design problem space”. *Design Studies*, *18*(4), pp. 441-445.

[6] Eisentraut R., Gunther J., 1997, “Individual styles of problem solving and their relation to representations in the design process”. *Design Studies*, *18*(4), pp. 369-383.

[7] Fricke, G., Successful approaches in dealing with differently precise design problems. Design Studies, 1999, 20(5), pp. 417-422.

[8] Goel, V. and Pirolli, P., The structure of design problem space. Cognitive Science, 1992, 16, pp395-429.

[9] Akin, O., and Chengtah, L., 1996, Analysing Design Activity, John Wiley & Sons, Chichester, UK, Design Protocol Data and Novel Design Decisions.

[10] Thomas, J. and Carroll, J., 1979, “The psychological study of design”. *Design Studies*, *1*(1), pp. 5-11.

[11] Cross, N. and Cross, A.C., 1998, “Expertise in engineering design”. *Research in Engineering Design*, *10*(3), pp. 141-149.

[12] Atman, C. J., Chimka, J. R., Bursic, K. M., and Nachtmann, H. L., 1999, “A Comparison of Freshman and Senior Engineering Design Processes”. *Design Studies*, *20*(2), pp. 131-152.

[13] Eisentraut, R., 1999, “Styles of problem solving and their influence on the design process”. *Design Studies*, *20*(5), pp. 431-437.

[14] Dorst, K. and Cross, N., 2001, “Creativity in design process: Co-evolution of problem–solution”. *Design Studies, 22*(5), pp. 425-437.

[15] Darke, J., 1979, “The primary generator and the design process”. *Design Studies, 1*(1), pp. 36-44.

[16] Valkenburg, R. and Dorst, K., 1998, “The reflective practice of design teams”. *Design Studies, 19*(3), pp. 249-271.

[17] Gero, J.S. and Kannengiesser, U., 2004, “The situated function-behaviour-structure framework”. *Design Studies, 25*(4), pp. 373-391.

[18] Kruger, C. and Cross, N., 2006, “Solution driven versus problem driven design: strategies and outcomes”. *Design Studies, 27*(5), pp. 527-548.

[19] Ho, C., 2001, “Some phenomena of problem decomposition strategy for design thinking: differences between novices and experts”. *Design Studies, 22*(1), pp. 27–45.

[20] Ball, L., Ormerod, T. and Morley, N., 2004, “Spontaneous analogising in engineering design: a comparative analysis of experts and novices”. *Design Studies, 25*(5), pp. 495-508.

[21] Gero, J.S. and McNeill, T., 1998, “An approach to the analysis of design protocols”. *Design Studies, 19*(1), pp. 21-61.

[22] Dinar, M., Shah, J.J., Langley, P., Hunt, G., Campana, E., 2011, “A structure for representing problem formulation in design”. In Proceedings of ICED11 (in press).

[23] Gero, J.S., Tham, K.W. and Lee, H.S., 1991, IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design (IntCAD91). Columbus, OH, USA, Behaviour: a link between function and structure in design

[24] Hirtz, J. M., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L., 2001. “Evolving a functional basis for engineering design”. In Proceedings of DETC01.

[25] ISO 10303-11:2004, 2004. Part 11: Description methods: The EXPRESS language reference manual.

[26] Dorst, K., 1996, Analysing Design Activity, pp17-34 (John Wiley & Sons, Chichester, UK, The Design Problem and its Structure.

[27] Bergsjo, D., Almefelt, L., Dinar, M. and Malmqvist, J., 2010, “Customizing Product Data Management for Systems Engineering in an Informal Lean-Influenced Organization,” *Systems Research Forum, 4*(1), pp.101-120.

[28] Baya, V., and Leifer, L., 1996. “Understanding information management in conceptual design”. In *Analysing Design Activity*, N. Cross, H. Christiaans, and K. Dorst, eds. John Wiley & Sons, ch. 7, pp. 151–168.

[29] Grebici, K., Aurisicchio, M., and Bracewell, R., 2009. “Guiding engineering design activities through a question based approach”. In Proceedings of DETC09.

[30] Novak, J. D., and Ca˜nas, A. J., 2008. The theory underlying concept maps and how to construct and use them. Tech. rep., Florida Institute for Human and Machine Cognition (IHMC).

[31] Newell, A., and Simon, H. A., 1972. *Human Problem Solving*. Prentice-Hall, Upper Saddle River, NJ.

[32] Quillian, R., 1966. “Semantic memory”. PhD thesis, Carnegie Institute of Technology, Pittsburgh, Pennsylvania, February.

