The Third International Conference on Design Creativity (3rd ICDC)
ISBN 978-1-904670-60-5
Bangalore, India, 12th-14th January 2015

TOWARDS A COMPREHENSIVE TEST OF PROBLEM FORMULATION SKILL IN DESIGN

M. Dinar[1], J. J. Shah[1], S. R. Todeti [2]

[1]Mechanical and Aerospace Engineering, Arizona State University, Tempe, USA
[2] Indian Institute of Science, Bangalore, India

Abstract: A battery of tests for assessing a number of cognitive skills relevant to conceptual design is being developed. Tests on divergent thinking (DT), visual thinking (VT), and qualitative reasoning (QR) are fully developed and validated. This paper focuses on the development of a test on the problem formulation (PF) skill. Indicators of problem formulation are identified and categorized as: insight, setting priorities, expanding, bounding, and structuring the design space. Unlike the previous tests which were developed and taken solely on pen and paper, this test can be developed with Problem Maps—data that is collected with the web-based Problem Formulator tool. A set of metrics for measuring skill levels are proposed with respect to which a class of graduate students have been graded. Candidate test items are discussed based on a number of design tasks which have been given to the students.

Keywords: *design skill tests, problem formulation, problem map*

1. Introduction

A successful designer should possess a variety of skills. A skill in this context is defined as a cognitive ability to perform an engineering design task. Traditional education of engineering design had focused on analytical skills. Shah (2005) identified a different set of skills with a focus on early stages of conceptual design. A battery of tests have been developed for measuring these skills: divergent thinking (Shah, Millsap, Woodward, & Smith, 2012), and visual thinking (Shah, Woodward, & Smith, 2013), qualitative reasoning (Khorshidi, Shah, & Woodward, 2013; Khorshidi, Woodward, & Shah, 2012). This paper discusses a preliminary structure for testing the remaining skill on the proposed set, problem formulation (PF). While the medium for developing and taking the first three tests was restricted to pen and paper, the PF test is set to take advantage of the Problem Formulator web-based tool (Maclellan, Langley, Shah, & Dinar, 2013) for tentative data collection and test taking.

2. Indicators of the PF skill

Problem formulation is the process of forming one’s understanding of a given problem. As Harfield (2007) argues, different designers form different understandings of a problem as ‘given’ and come up with different solutions to different problems (perceived by designers when reading the same problem statement). This section discusses the different components of the PF skill and proposes the corresponding metrics for evaluating each subskill.

2.1. PF subskills

To categorize the components of the PF skill, one possibility is to see how designers use knowledge. Kruger and Cross (2006) used an expertise model which consisted of application knowledge (inference, and domain knowledge) and problem solving knowledge (methods and strategies). One can use this classification in problem formulation to state that designers use *insight* and *set priorities* (application knowledge) while they *expand*, *bound*, and *structure* the design space (problem solving knowledge) in the process. PF subskills are explained with regards to the highlighted categories. But first, two points should be clarified: it is known that during design, problem and solutions spaces co-evolve (Dorst & Cross, 2001; Maher & Tang, 2003); in the same manner, it is difficult to draw a line between the problem space and the solution space, therefore the term design space is preferred to either problem or solution spaces.

2.1.1. Insight

A design problem often starts with a problem statement where some customer needs are explicitly stated. The designer must then discover implicit requirements that are necessary to meet. These implicit requirements can be additional requirements at the top level, or derived, as existing requirements are decomposed further. Checklists have been provided for eliciting requirements (Pahl & Beitz, 1996). This subskill is *requirement elicitation*.

As mentioned above, working on design problems involves elements of problem and solution spaces, e.g., when defining a requirement a designer may have to consider parametric equations governing the use environment or a specific part. *Relationship identification* among different aspects of the problem is another subskill involving insight.

Design problems are ill-defined (Simon, 1973). Observations of designers in different studies include active search for sources of information other than the problem statement (Cross, Dorst, & Roozenburg, 1992; Eris, 2004). This is the *information seeking* subskill.

All the information that designers seek and acquire is not helpful; it can even be misleading. One of the differences between novice and expert designers is that while experts filter unnecessary information, novices try to make something out of everything (Atman, Chimka, Bursic, & Nachtmann, 1999; Kim, Kim, Lee, & Park, 2007). This is called here *relevance detection*.

One of the causes of bad designs is that designers fail to consider who uses the end product and how. Norman (1990) gives many such examples in his affordance-based design approach. One can conclude that an account of insight in formulating a design problem is the ability to identify use scenarios, or *use description*.

2.1.2. Setting priorities

One characteristic of formulating a design problem is to understand where one should pay the main attention to, as resources are limited in a design project. One of the main differences between experts and novices is that experts quickly identify and focus on the most critical issues, while novices treat everything equally (Ho, 2001). This is *key issue identification*.

Similarly, successful designers apply their knowledge to distinguish between real and fictitious constraints (Shah, 2005). The addition of such requirements unnecessarily restricts the design space that is being explored resulting in a decrease in the creativity of the outcome (Maher, Poon, & Boulanger, 1996). This is called here *fictitious problem avoidance*.

2.1.3. Expanding design space

Maher et al. (1996) discuss the processes of the early stages of conceptual design and find a distinction between exploring the design space, and searching the design space. While the former concerns the addition of new dimensions in the space, the latter is about finding possible points in that

dimension. It is difficult to draw such a strict line in the formulation data without having the designers explain the cognitive process they go through. However, for the sake of the PF test, one can assume that exploration relates to using domain knowledge to create a new PF data fragment and thus *insight* while search is an expansion of the existing fragments. One can track designers' PF to see how they use their existing formulation; will they look for analogues, will they look for similar cases? Ball, Ormerod, and Morley (2004) have found that experts lean on experiential abstract knowledge while novices rely on case-driven analogies, mainly driven by surface-level cues. Abstraction can foster creativity in early conceptual design (Ward, Patterson, & Sifonis, 2004). Hence *problem abstraction*.
One of the aspects that makes good designers stand out is the ability to deliver surprising features in the design that delights customers. The well-known Kano model (Kano, Seraku, Takahashi, & Tsuji, 1984) differentiates between basic features and features of delight in a design where the mere presence of the latter increases customer satisfaction. *Delight addition* thus can be another skill in expanding the design space.

2.1.4. Bounding design space

In the same way that problem and solution spaces co-evolve during design and cannot be separated, PF skills involve convergence in addition to divergence. An aspect of defining the problem is *specification*, setting the boundaries of variables, constraints, etc. QFD (Kogure & Akao, 1983) and creating spec sheets (Pahl & Beitz, 1996) are established methods in this regard.
When formulating a problem, designers also have to start narrowing the scope at a reasonable time. Comparing freshmen and senior designers, Atman et al. (1999) found that spending a large proportion of time on defining the problem did not necessarily result in quality designs for freshman, and seniors' problem *scoping* resulted in better designs.

2.1.5. Structuring design space

Designers not only expand and bound the design space during problem formulation, but also structure the space. One of the differences between experts and novices is that experts excel at explicitly decomposing the problem the problem (Ho, 2001). Gero and Mc Neill (1998) also found *decomposition* as one of macro strategies that designers adopt in early stages of conceptual design.
In structuring the design space, designers employ different representations with different modalities, e.g., [visual] sketches, [verbal] conversations, [tactile] prototypes. Akin, Chengtah, and Lin (1996) found that novel decisions occur in multi-modal episodes of designing. Therefore, having different means of *representation* is a subskill in formulating design problems.

2.2. PF metrics

Metrics can be defined for measuring the PF skills that were explained above. The number of added requirements that are necessary to achieve but implicit, i.e., not directly mentioned in the problem statement, can indicate *requirement elicitation*. The number of identified relations between different fragments of the problem can be a measure of the *relationship identification* skill. The number of times that a designer requests additional information that are important in the design and not apparent in the problem statement, or refers to external sources of information that are known to the designer are indicators of *information seeking*. If the problem statement has unnecessary, redundant, or misleading information, *relevance detection* can be measured by the degree to which this information is omitted. *Use description* can be measured by the number of times the designer identifies pertinent environmental variables or user affordances. The number of identified key issues and the degree to which the designer allocates resources to them can measure *key issue identification*. The number of added fictitious requirements can measure the *fictitious problem avoidance* skill. One should note that if this measure is intended to be reported on a similar scale with other measures, the inverse or complement of this number should be used, i.e., better designers have fewer fictitious requirements.
The number of times a designer abstracts a fragment rather than specifying it can be a measure of *problem abstraction*. The number of auxiliary features of delight that are added can indicate *delight addition*. The portion of parameters that are bounded with absolute or relative ranges and targets

constitute a measure of *specification*. *Scoping* may be measured by the degree to which the direction of decomposing the problem shows honing in on a framed problem. The level of decomposing different aspects of the problem, e.g., the depth of an objective tree or number of disjunctive functional decompositions, can be indicators of the *decomposition* skill. Finally, *representation* can be measured with the number of times a designer resorts to different types of representing aspects of the problem, e.g. sketches, calculations, simulations, or physical prototypes. Table 1 summarizes the PF skills, their subskills, and the corresponding metrics for measuring each subskill.

Table 1. PF subskills and measures

Skill	Subskill	Definition	Metric
Insight	Requirement elicitation	Ability to identify implicit necessary objectives, requirements, constraints	Number of added necessary requirements
	Relationship identification	Ability to identify relationships between variables	Identified relations between different parts of the problem
	Information seeking	Ability to identify missing information	Requests of and references to additional information
	Relevance detection	Ability to identify relevant and irrelevant information	Omission of unnecessary information presented in the problem statement
	Use description	Ability to identify use scenarios	Identified environmental variables and user affordances
Setting priorities	Key issue identification	Ability to identify key issues	Level of attention to the key issue
	Fictitious problem avoidance	Ability to distinguish between real and fictitious constraints	Number of fictitious requirements
Expanding design space	Problem abstraction	Ability to map a problem to a higher dimension	Number of abstractions
	Delight addition	Ability to add surprise and delight	Number of auxiliary delightful features
Bounding design space	Specification	Ability to set constraints, quantities, parameter ranges	Identified ranges and targets
	Scoping	Ability to scope the problem space	Direction of decomposing aspects of the problem
Structuring design space	Decomposition	Ability to decompose problems, functions and artifacts	Level of decomposing aspects of the problem
	Representation	Ability to build multiple representations	Level of implementing sketches, simulations, prototypes

3. Implementing Problem Maps for measuring the PF skill

3.1. The Problem Map framework

The Problem Map framework (P-maps) is a fine-grained ontological framework created for understanding the differences among designers in problem formulation, in addition to improving the process of data collection and analysis. It has five types of entities: *Requirement*, *Function*, *Artifact*, *Behavior*, and *Issue*. Each type consists of entities whose instances can be a part of disjunctive hierarchies. Entities can have optional attributes. The entities in the five groups can form parent-child relations. All groups are inter-related with bidirectional relations; Issues can have a relation to any combination of the rest of the entities.

Requirements describe the specifications of the design problem. Functions refer to the actions that the design should execute to accomplish its requirements. Artifacts describe the physical components of the design or the concepts the design may be using. Behaviors are the physical properties and laws that govern the design in terms of equations and physical effects, as well as the parameters that are relevant to both artifacts and functions. Issues describe the problems associated with other entities in the design formulation. An issue is often an expression of a point that the designer believes to be pivotal or problematic in achieving a design objective. To name a few, an issue can arise in realizing a function with a specific artifact or behavior, in realizing conflicting design goals such as lower weight and strength of a structure, or in accommodating different components in a product architecture due to incompatible interfaces.

Based on P-maps, the interactive web-based tool Problem Formulator tool has been created (Maclellan et al., 2013), see Figure 1. Though users have used the tool for documenting and representing problem formulation of complete design tasks (class or capstone projects), the main objective of the tool is to serve as a testbed for collecting massive amounts of data. This will include data from responses to bounded questions in controlled settings (scope of the problem, allotted time, and location of accessing the tool) that will be devised for the PF skill test.

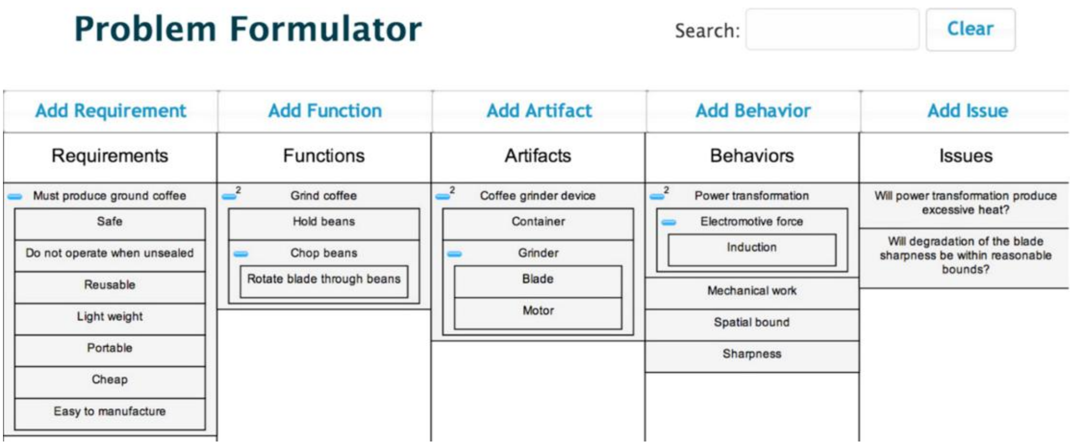

Figure 1. A screen shot of the Problem Formulator tool (Maclellan et al., 2013)

3.2. Measuring PF with P-maps

The P-maps ontology provides various types of analyses in terms of temporal changes among states at different times, or measures of a state at a specific time. For example, one can trace whether a designer abstracts a function, i.e., add a parent function to a child node instead of decomposing it further to other children (temporal change); or count the number of derived requirements (state measure). Table 2 proposes a set of P-map measures for the PF subskills. This relation only shows the corresponding measures that one can calculate from a P-map; it does not specify a scoring or grading schema. Scoring the skills can be based on comparing test takers' responses to a normative P-map for the given question. For example, for scoring the *key issue identification* subskill one can create an aggregate of possible issues for the given problem and assign the highest score when the test taker includes all the issues on the list in his P-map, and proportionally lower scores for fewer issues. As can be seen, *representation* does not have a P-map measure since the data modality in the existing tool is limited to text.

3.3. Creating a grading schema

P-maps of three design tasks collected from a class of twenty one graduate students of mechanical design were used for evaluating their PF skills objectively. An aggregate of all responses formed different inventories such as relevant and fictitious requirements or key issues which were the basis for comparing students to each other. For each subskill, a range was defined for good, fair, and poor scores, see Table 3. For example, for *relationship identification*, if there were more than one parametric relation under Behaviors and more than one relation among groups of entities, the score would be five points; three points were given if one of the relation types was not shown; and 1 point for no identified relations. This is an ongoing work and all proposed measures have not been graded yet. Figure 2 shows the results of grading the students for the design task whose grading schema was discussed above.

Table 2. P-map measures for PF subskills

Subskill	**P-map measure**
Requirement elicitation	Derived lower level Requirements; Added top level Requirements
Relationship identification	Parametric relations in Behaviors; Inter-group relations to Behavior
Information seeking	Issues related to Requirements and Behaviors
Relevance detection	Irrelevant given [undeleted] Requirements
Use description	Environment derived Requirements; Behaviors in relation to use Requirements
Key issue identification	Key Issues
Fictitious problem avoidance	Irrelevant derived Requirements
Problem abstraction	Abstracted Requirements, Functions, Artifacts, Behaviors
Delight addition	Novel Artifacts; Derived performance Requirements
Specification	Behavior parameters with ranges; Behaviors related to Requirements
Scoping	Top-down ordered Functions
Decomposition	Requirement depth; Function depth; Artifact depth

Table 3. A grading schema based on P-maps for evaluating students' PF in a design task

Subskill	**Good (5 points)**	**Fair (3 points)**	**Poor (1 point)**
Requirement elicitation	Derived requirements>2	Derived requirements =1	No derived requirements
Relationship identification	Behavior parametric relation & intergroup relations >1	One relation type only >1	No relations
Relevance detection	Irrelevant requirements =0	Irrelevant requirements = 1	Irrelevant requirements>1
Key issue identification	Key issues > 1	One key issue	No key issues
Fictitious problem avoidance	Fictitious requirements=0	Fictitious requirements=1	Fictitious requirements>1
Problem abstraction	Abstraction >1	Abstraction=1	No abstraction
Delight addition	Novel artifacts>2	Novel artifacts =2	No novel artifacts
Decomposition	Function depth>2	Function depth =2	Function depth =1

4. Candidate test items

The example shown in the previous section was for a complete design task. Questions in a test can be in a more controlled setting. This section gives examples of possible questions for different parts of the test. Similarly to the previous example, aggregate of responses can be turned into inventories for comparison of the test takers' responses to the norm. One of the important characteristics of a test is how well it reflects on differences among takers. In order to have a test with an appropriate distribution, the questions should be balanced, i.e., most subjects should be able to answer easy questions; some subjects respond better to more difficult questions; a few find the most difficult answers. For example, one of the design tasks in the class was design of a device that collects Lego blocks and disposes of them in a bin through a ramp. One of the requirements that some students did not elicit was that a clearance was needed at the base of the ramp for a vehicle. Failure to finding this requirement resulted in some final designs getting stuck at the base, see Figure 3.

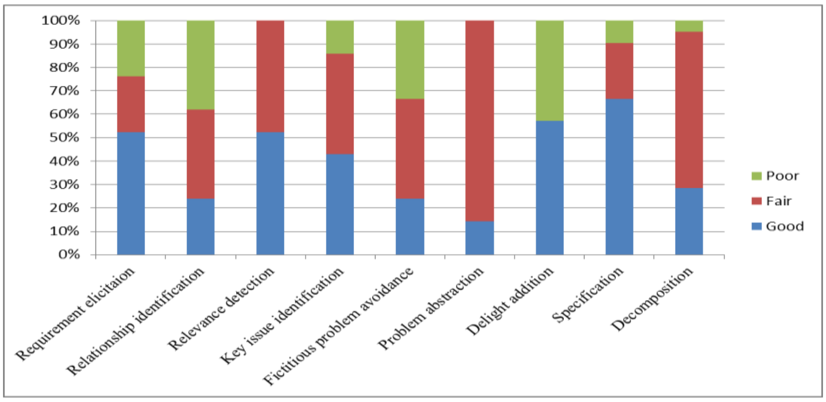

Figure 2. Distribution of students' grades of PF skills for a design task in a class

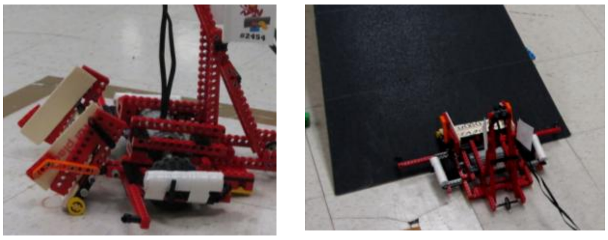

Figure 3. An example of a failed design because of missing an implicit requirement

Devising some questions requires more elaborate framing. For example to test scoping one can give the following setting. Consider a function decomposition with 4-5 branches and 3-4 levels deep. A list of 8-10 of these functions from different levels and branches, but including all functions of one of the branches is shown to the subject. An indirect hint is given about scoping, e.g. that the subject is about to make preliminary analysis of a design and they should pick 4 functions from the list. If they pick the 4 functions of that one branch, it can be a sign of good scoping. If they are asked to pick the functions in a specific order, we can also measure the abstraction sub-skill too. Testing multiple skills in one question is used in validating tests.

5. Discussion

5.1. Test development

Unlike the previous design skill tests, this test is planned to be taken on a web-tool, not pen and paper with the intention that more subjects take the test and be graded quickly. One remaining challenge in using the tool for this purpose is that some automatic text processing is required for assessing the free form text responses collected in P-maps. Inclusion of counterfactuals in responses is another difference in this test. As shown earlier some measures such as fictitious requirements negatively affect scores. It will be difficult to know whether subjects avoided fictitious requirements deliberately or out of luck.

5.2. Test validation

For repeatability, multiple questions can measure each subskill and vice versa. The tool can help with this regard, as some of its functionalities, e.g., relating groups of entities can be used for each question. With further development of the test, other methods such as factor analysis and correlation studies can be used to find overlapping or redundant skills that can be merged together or omitted, similarly to what has been done with previous parts of the design skill tests (Shah et al., 2012, 2013).

Acknowledgement

This study is supported by CMMI Grant Number 1002910 from the National Science Foundation. The opinions expressed in this paper are those of the authors and not endorsed by the NSF.

References

Akin, Ö., Chengtah, L., & Lin, C. (1996). Design protocol data and novel design decisions. In N. Cross, H. Christiaans, & K. Dorst (Eds.), *Analysing Design Activity* (Vol. 16, pp. 35–63). Chichester, UK: John Wiley & Sons.

Atman, C. J., Chimka, J. R., Bursic, K. M., & Nachtmann, H. L. (1999). A Comparison of Freshman and Senior Engineering Design Processes. *Design Studies*, *20*(2), 131–152.

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

Cross, N., Dorst, K., & Roozenburg, N. (1992). *Research in design thinking: proceedings of a workshop meeting held at the Faculty of Industrial Design Engineering, Delft University of Technology, the Netherlands, May 29-31, 1991* (p. 215). Delft: Delft University Press.

Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem–solution. *Design Studies*, *22*(5), 425–437.

Eris, O. (2004). *Effective inquiry for innovative engineering design* (p. 154). Boston: Kluwer Academic Publishers.

Gero, J. S., & Mc Neill, T. (1998). An approach to the analysis of design protocols. *Design Studies*, *19*(1), 21–61.

Harfield, S. (2007). On design "problematization": Theorising differences in designed outcomes. *Design Studies*, *28*(2), 159–173.

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

Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). Attractive quality and must be quality. *Quality*, *14*(2), 39–48.

Khorshidi, M., Shah, J. J., & Woodward, J. (2013). Rethinking the Comprehensive Test on Qualitative Reasoning for Designers. In *Proceedings of ASME IDETC/CIE* (p. V005T06A027). Portland, OR, USA.

Khorshidi, M., Woodward, J., & Shah, J. J. (2012). Towards a Comprehensive Test of Qualitative Reasoning Skill in Design. In *Proceedings of ASME IDETC/CIE* (p. 889). Chicago, IL, USA: ASME.

Kim, M. H. H., Kim, Y. S. S., Lee, H. S. S., & Park, J. a. A. (2007). An underlying cognitive aspect of design creativity: Limited Commitment Mode control strategy. *Design Studies*, *28*(6), 585–604.

Kogure, M., & Akao, Y. (1983). Quality function deployment and CWQC in Japan. *Quality Progress*, *16*(10), 25–29.

Kruger, C., & Cross, N. (2006). Solution driven versus problem driven design: strategies and outcomes. *Design Studies*, *27*(5), 527–548.

Maclellan, C. J., Langley, P., Shah, J. J., & Dinar, M. (2013). A Computational Aid for Problem Formulation in Early Conceptual Design. *Journal of Computing and Information Science in Engineering*, *13*(3), 031005.

Maher, M. Lou, Poon, J., & Boulanger, S. (1996). Formalising Design Exploration as Co-Evolution: A Combined Gene Approach. In *Advances in Formal Design Methods for CAD: Proceedings of the IFIP WG5.2 Workshop on Formal Design Methods for Computer-Aided Design*. London, UK: Chapman & Hall.

Maher, M. Lou, & Tang, H. (2003). Co-evolution as a computational and cognitive model of design. *Research in Engineering Design*, *14*(1), 47–63.

Norman, D. A. (1990). *The design of everyday things*. New York: Currency, Doubleday.

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

Shah, J. J. (2005). Identification, Measurement and Development of Design Skills in Engineering Education. In A. Samuel & W. Lewis (Eds.), *Proceedings of the 15th International Conference on Engineering Design (ICED05)* (p. DS35_557.1). Melbourne, Australia.

Shah, J. J., Millsap, R. E., Woodward, J., & Smith, S. M. (2012). Applied Tests of Design Skills — Part 1: Divergent Thinking. *Journal of Mechanical Design*, *134*(2), 021005.

Shah, J. J., Woodward, J., & Smith, S. M. (2013). Applied Tests of Design Skills—Part II: Visual Thinking. *Journal of Mechanical Design*, *135*(7), 71004.

Simon, H. A. (1973). The structure of ill structured problems. *Artificial Intelligence*, *4*(3-4), 181–201.

Ward, T. B., Patterson, M. J., & Sifonis, C. M. (2004). The Role of Specificity and Abstraction in Creative Idea Generation. *Creativity Research Journal*, *16*(1), 1–9.