DOI:10.1145/2926712

Orit Hazzan and Jeff Kramer

Viewpoint
Assessing Abstraction Skills

What makes a good question?

IMAGE BY SJOERD LOS/FLICKR

ABSTRACTION IS CONSIDERED to be a key skill underlying most activities in computer science (CS) and software engineering (SE).[3,4] As a central concept, abstraction is taught and utilized in various guises in every CS and SE course: in requirements specifications, problem solving, and modeling through to programming and debugging. Given its importance, can one assess an individual's abstraction ability? Are abstraction skills assessable at all? If no, why? If yes, how? We decided to conduct a survey to investigate this topic. As far as we know, this is one of the first trials to address this challenge.[1]

In 2007, Kramer proposed to develop a test that could assess abstraction skills in the context of CS and SE, and which would be more multifaceted than that used in psychometric tests.[4] In this spirit, we decided to consult experts in CS and SE research and teaching concerning the suitability of various question patterns (or templates) for assessing abstraction skills. We specifically used patterns rather than specific questions in order not to limit the experts' line of thought, and, at the same time, to provide a template that each instructor could adjust and populate according to his or her needs.

Our data analysis reveals that expert instructors tend to agree about the suitability of a pattern for checking abstraction ability when it asks students to construct the abstraction rather than when applying abstraction principles. We explain this approach by the constructionism learning theory of Papert and Harel.[5]

Survey Structure

We argue that directly asking experts the question "Are abstraction skills assessable?" would be unlikely to elicit a clear indication of how to go about constructing such an assessment tool. As a concrete proposal, we therefore developed a set of 10 patterns of questions and asked the experts to assess each pattern on a 1–10 scale according to its suitability for assessing abstraction skills. In addition, for each pattern we presented three open questions regarding:

- What specific abstraction skills does Pattern X measure?
- Can you suggest an example that best fits Pattern X?
- Any additional comments on Pattern X?

In this way, we gained both a quantitative and qualitative perspective of the experts' opinion.

Table 1. Survey participants' demographic data.

Where do the experts teach?		
Research university	23	74%
Teaching college	5	16%
High school	0	0%
Other*	3	10%
	31	

* One indicated that he or she is a post doc who taught in the past and one indicated that she or he works in a research institute.

Teaching Experience		
0–5 years	2	6%
6–10 years	2	6%
11–15 years	3	10%
16–20 years	3	10%
More than 20 years	21	68%

Students' major/minor CS/SE taught by the experts (more than one option can be checked)	
Computer science major students	25
Computer science minor students	10
Prospective computer science high school teachers	1
Software engineering students	18
Electrical engineering students	5
Other engineering students	9
Other	7

Courses the experts teach (more than one option can be checked)	
Introductory computer science courses	18
Advanced computer science courses	20
Software engineering courses	21
Engineering courses	5
Other	6

Survey Participation

We approached two groups with similar characteristics: 20 experts were approached based on personal relationships and 11 responded to a message that we sent to a mailing list.[a] N = 31 is a reasonable number for such a preliminary analysis. The demographic data presented in Table 1 indicates that the majority of the experts work at research universities, have many years of teaching experience of both introductory and advanced CS and SE courses to both CS major and SE students. This background clearly indicates they understand issues related to the challenge of teaching the topic of abstraction to good CS and SE students as well as assessing their abstraction skills.

Data Analysis: When Do CS and SE Instructors Agree and Disagree about the Appropriate Means to Assess Abstraction?

For each pattern, we calculated the average evaluation score and the standard deviation (see Table 2). Though an agreement was not reached with respect to any pattern, we can see that:

- The *highest agreement* (Average = 7.53; SD = 2; seven different scores) with respect to a pattern's suitability for measuring abstraction skills was reached for Pattern 5: Given a system representation, students are asked to give one representation that is more abstract than the given one and one representation that is less abstract than the given one.
- The *lowest agreement* (Average = 6, SD = 3.05; 10 different scores) with respect to a pattern's suitability for measuring abstraction skills was reached for Pattern 2: Given several representations of a specific system, students are asked to rank them according to their level of abstraction and to consider the purpose of each abstraction.

Disagreement: What was surprising in the data analysis was the vast range

[a] From the questionnaire sent to the first group (http://bit.ly/2f9GWgr), two questions were excluded in the subsequent questionnaire sent to the second group (http://bit.ly/2eQJH7N): a) Question 1 was a specific case of question 2; and b) Question 9 that the first group assessed lowest concerning its ability to assess abstraction skills. In addition, to the second questionnaire, we added the open question "Can you suggest an example which best fits Pattern X?"

Table 2. CS and SE instructors' assessment of the survey patterns.

	Pattern 1	Pattern 2	Pattern 3	Pattern 4	Pattern 5	Pattern 6	Pattern 7	Pattern 8	Pattern 9	Pattern 10
Average	6.26316	6	6.464286	7.285714	7.535714	6.518519	6.785714	6.666667	4.789474	6.1923077
SD	2.90291	3.05505	2.545616	2.565543	2.008909	2.517177	2.424522	2.557042	2.323287	3.2163223
N	19	28	28	28	28	28	28	27	19	27
# of scores	10	10	10	8	7	8	9	9	8	9

of scores and disagreement exhibited with respect to almost all patterns concerning their suitability to assess abstraction skills. For example, patterns 1–3 received all scores. These three patterns provided the students with a set of object representations and asked the students to categorize them according to the student's choice of abstraction. In other words, in these patterns, students were *not* asked to create/construct an object while considering abstraction.

Agreement: As mentioned, Pattern 5 has the highest average score with the lowest distribution. In this pattern and in another one (Pattern 4, which also was assessed high), the students were asked to *create some system representations by referring to abstraction*. We may conclude that instructors tend to agree that constructing abstraction is a suitable way to assess abstraction skills. One of our experts explains: "This is a nice concrete type of pattern question. Looks at abstraction generation skill but in a much more constrained setting."

Psychological Explanation

Theoretically, we explain the experts' ranking by the theory of constructionism, inspired by the constructivist theory. According to constructivism, learners construct mental models in order to understand the world around them. Constructivism advocates student-centered, discovery learning. One outcome of the constructivism framework is that one has to be actively involved with the material he or she learns.[2] Accordingly, "The simplest definition of constructionism evokes the idea of learning by making … .[5]

The scores presented in Table 2 indicate that the experts, who answered our questionnaire, saw the potential contribution of Pattern 5 to students' understanding. From a constructionism perspective, this can be explained by the fact that Pattern 5 asks students to *construct* two representations—a less and a more abstract one than a given one, rather than to categorize, compare, or explain some aspect of a given object while referring to abstraction. Here are several instructors' answers to the question: "What specific abstraction skills does Pattern 5 measure?" which reflect this explanation [italics were added by the authors]:

- Ability to *invent* new abstractions at different levels of abstraction.
- Ability to *"do"* abstraction—to throw away detail, while keeping the essential structure. Also, ability to "*do*" refinement—to add detail.
- Ability to *CREATE* (rather than just assess), from an external knowledge source, different levels of abstraction.
- The ability to *maneuver* between abstraction levels, as needed, by adding/removing details. Also, to identify the entities that comprise an abstraction level, and, constructively, detailing or abstracting them in order to achieve the desired level of abstraction.
- … the ability to *devise* abstractions at various levels.

Conclusion

The question: "Are abstraction skills assessable?" is part of the discussion on teaching soft CS concepts. The experts' responses, which reflect a constructionism perspective, indicate that they value students' active engagement and creativity in the learning process. One of the instructors suggested that Pattern 5 "… seems too open ended to me for a test question—although great for a studio-style discussion."

Based on the experts' answer, this Viewpoint has focused on distinguishing between the creative construction of abstractions as against the passive assessment of abstraction level. Another particularly crucial aspect in abstraction is that of purpose and the consideration of fit-for-purpose in activities such as modeling and problem solving. This includes both the creation of new abstract formalisms and languages to express abstractions—as is created by researchers and language designers, and the use of these to create specific abstractions for specific applications. We believe that abstraction purpose needs to be more directly and explicitly considered and discussed by instructors, students, and practitioners when reflecting on their use of abstraction. In this spirit, one of the experts who responded to the survey suggested to highlight "[…] what the system needs to achieve and what are the criteria for better solution. It will improve the quality of the answers and will provide wider range of good answers." Another one added: "I would recommend to couple it with a specific purpose to guide what kind of aspects that can be left out or needs to be included."

We see this Viewpoint as a starting point of the discussion of this topic and invite the ACM community to join and contribute to the discussion. Such a discussion is especially relevant today when a variety of teaching methods and frameworks are available for students who seek the highest return for their effort investment.

References

1. Bennedsen, J.B. and Caspersen, M.E. Abstraction ability as an indicator of success for learning object-oriented programming? *SIGCSE Bulletin inroads 38*, 2 (2006), 39–43.
2. Davis, R., Maher, C., and Noddings, N. Introduction: Constructivist views on the teaching and learning of mathematics. In R. Davis, C. Maher, and N. Noddings, Eds. *Constructivist Views on the Teaching and Learning of Mathematics.* National Council of Teachers of Mathematics, Reston, VA, 1990, 7–18.
3. Denning, P.J., et al. Computing as a discipline. *Commun. ACM 32*, 1 (Jan. 1989), 9–23.
4. Kramer, J. Is abstraction the key to computing? *Commun. ACM 50*, 4 (Apr. 2007), 37–42.
5. Papert, S. and Harel, I. *Constructionism.* Ablex Publishing Corporation, 1991.

Orit Hazzan (oritha@technion.ac.il) is a Professor of the Faculty of Education in Science and Technology at Technion–Israel Institute of Technology.

Jeff Kramer (j.kramer@imperial.ac.uk) is Professor of Distributed Computing in the Department of Computing at Imperial College London.

We would like to thank the experts who answered our survey, invested the time, and shared their thoughts with us.