A Simulation Task to Assess Students' Design Process Skill

Judith E. Sims-Knight[1], Richard L. Upchurch[2], and Paul Fortier[3]

Abstract - **Research has shown that the quality of one's design process is an important ingredient in expertise. Assessing design process skill typically requires a performance assessment in which students are observed (either directly or by videotape) completing a design and assessed using an objective scoring system. This is impractical in course-based assessment. As an alternative, we developed a computer-based simulation task, in which the student respondent "watches" a team develop a design (in this instance a software design) and makes recommendations as to how they should proceed. The specific issues assessed by the simulation were drawn from the research literature. For each issue the student is asked to describe, in words, what the team should do next and then asked to choose among alternatives that the "team" has generated. Thus, the task can be scored qualitatively and quantitatively. The paper describes the task and its uses in course-based assessment.**

Index Terms - Design Process, Assessment, Continuous Improvement, Simulation

INTRODUCTION

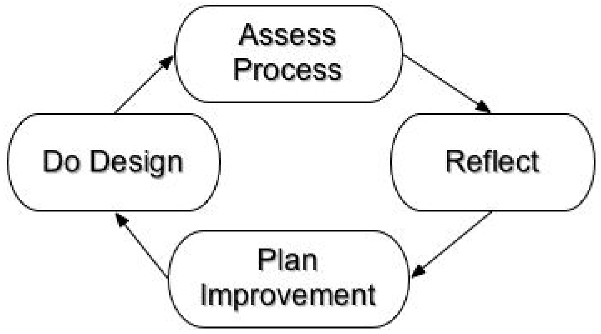
Research on the development of expertise has shown that, no matter what the field, it takes at least ten years of extensive practice to reach the top of one's field. This is true in every field studied—science, chess, music, art, etc. [1]. This research also clearly indicates that such extensive practice is a necessary, but not sufficient condition for the development of expertise. Ericsson [1] argues that in addition deliberate practice, practice in order to improve, is necessary.

Expertise in engineering includes both analytic and synthetic skills and, to become a successful practitioner requires that novices get extensive practice in both skill sets. Many engineering educators have realized that their curricula are heavily weighted toward analysis and the result has been a move toward senior design courses in which students engage in open-ended design and development of a substantial product. Now it is becoming clear that such curricular innovations, while a step in the right direction, are not sufficient to develop expertise in creating open-ended designs.

Although educators can not hope to produce expert designers in four years, they can hope to lay the foundations for students to develop that expertise when they graduate. To do so requires that they help students to develop habits of deliberate practice. We argue that there are two components to deliberate practice in engineering design. One component is understanding of and development of skill in design process, the second is reflective practice.

Research on design process has demonstrated that although quality of process correlates with quality of product [2], excellent process does not ensure excellent product and v. v. [3]-[6]. Thus, excellent designers must have both design and design process expertise. Unfortunately, it is not easy to capture experts' design process, because much of experts' knowledge is intuitive and thus difficult to describe. Luckily, researchers of design realized this many years ago and have been working hard to find ways of characterizing excellent design process. One of the first discoveries was that the principles abstracted by prescriptive analyses extant in the field were not necessarily the principles actually used by expert practitioners [7]. Thus, an approach that asks students to learn prescriptive principles and then apply those principles in open-ended design will fail. It is important, therefore, to base the teaching of design process on what actually works for practitioners. Furthermore, educational research has made it clear that people learn through action and not through direct tuition [8].

The second component needed to help students develop habits of deliberate practice is reflective practice. Students need to become good practitioners of personal continuous improvement (see Figure 1). They need to be able to assess their performance on a design task, both with respect to the product and the process, to reflect on that assessment, and to figure out ways to improve.



The goal of our current NSF-supported project has been to develop a series of tools that assess the various aspects of design process knowledge. These tools are designed to be

[1] Judith E. Sims-Knight, Psychology, University of Massachusetts Dartmouth, jsimsknight@umassd.edu
[2] Richard L. Upchurch, Computer and Information Science, University of Massachusetts Dartmouth, rupchurch@umassd.edu
[3] Paul Fortier, Electrical and Computer Engineering, University of Massachusetts Dartmouth, pfortier@umassd.edu

embedded in a course-based continuous improvement process that can be used both by instructors for course improvement and by students for individual process skills improvement. To promote improvement in design process knowledge, an assessment tool must both tap the important components of knowledge and provide pointers toward improvement.

In this project we have developed assessments for all components of design process knowledge—declarative knowledge [9], structure of knowledge [10], and use-in-action [11]. This paper describes our design process simulation task, which isolates the use-in-action of design process from actual designing. The next section will describe our task analysis and the following section describes the simulation task.

CHARACTERISTICS OF EXPERT DESIGN PROCESS

It is extremely difficult to study the cognitive processes that underlie expert design because the processes are invisible to both the observer and to designers themselves. The standard way of collecting data on design process is to find design professionals varying in years of experience and asking them to think out loud while creating a design. The procedure is typically videotaped and transcribed and the responses coded into various categories. Independent coding by two individuals is compared to assure that the coding is reliable. There is enough research on effective design process to guide instructional effort. Domains included in our analysis are software, electrical engineering, and mechanical engineering design. This research makes clear that experts, but not novices, attend to process explicitly and explicitly discuss what they plan to do next [12]-[13]. Professional designers (1) take time to understand the problem [13]-[15], (2) test frequently (e. g., check that understanding with techniques such as scenarios, checking preliminary design against requirements, etc. [13], (3) create alternative conceptual designs from the beginning [16]-[17] or, if using a one-design-at-a-time strategy, reconsider early designs at the first sign of trouble [18], (4) consider feasibility and tradeoffs, (5) use both top-down and opportunistic design when appropriate [16],[19]-[21], (6) consider interrelations among components [4], (7) review and reflect on design in terms of requirements [6],[20]-[21], and (8) prototype [21]-[22].

WHY A SIMULATION TASK

One way to observe students' design process might be to observe them creating a design. Although this has great face validity, it pales in the execution. First and foremost, students do not exhibit much process awareness, even when asked explicitly to include process information or to reflect on their process after completion of a design task [11]-[12] (a phenomenon that has been demonstrated in learning physics and other cognitive tasks [23]-[25]. Second, using simulations in other domains have been shown to be effective learning devices (see [26]-[28] for examples in the closely related domain of software management). Third, a simulation task can separate the cognitive tasks of actually creating a design from the cognitive tasks of thinking through process issues. Because both of these tasks tend to be cognitively overwhelming to students, focusing on one will reduce the overload. Fourth, administration time is reduced because the student does not need to generate the design. Fifth, it is a situated cognition task, so can promote authentic analysis of design process.

THE SIMULATION TASK

The simulation task presents a software design task, although it could easily be adapted to other kinds of design problems. The conceit is that the respondent is advising a design team comprised of four members, one who is team leader (Pedro). A fifth character, the manager (Tom), intrudes on occasion. The task is presented via the web.

The format of the simulation design follows a standard sequence. The respondent is told about what the team has done and presented the dilemma of what to do next. She or he first answers the question in narrative form. Then the program presents, in multiple-choice format, possible courses of action that the "team" has generated and the respondent has to choose among them. The number of alternatives varies from 2 to 5. Frequently there is more than one correct answer, which may be equally good or not. If the respondent chooses a nonoptimal response, the respondent is asked to explain his or her reasoning and then the program asks him or her to choose among the remaining options.

The multiple choice alternatives are all allocated a specific number of points ranging from –1 to +4. The -1 score is assigned to alternatives that will lead to negative outcomes and 0 to alternatives that will neither hurt nor help. The scoring scheme gives the highest score to students who choose the best alternative the first time they are given the choice and a lower score to those who figure it out on the second or third pass.

The numeric scores may be added to create an overall score. In addition, each individual answer, both open-ended and multiple-choice, can be analyzed to provide pointers toward improvement.

Table 1 lists the phases of design covered in the simulation task, the main points of each phase, and the design process principles exemplified. Although the sequence of events in the simulation follows a waterfall model, the focus of the simulation is on assessing students' understanding of process-guided decisions that exist in all models. Students are not asked questions to assess their understanding of a waterfall model.

The table does not, however, give an accurate view of the task itself. We will therefore describe the choice points of each phase. For the first phase, understanding the problem, we will describe the details as well. The full set of storyboards (see Figs. 1-3) is available at http://www.cogsci.umassd.edu/simtask/storyboard/figures/index.html.

I. Understanding the Problem

Figure 1 delineates the flow through the first phase and the scores assigned to each answer. The problem statement is presented and respondents are asked what the team should do first. Regardless of the answer, the program asks respondents to choose among the following alternatives the team has generated: “elaborate the problem statement” (0), “meet with the customer” (+4), or “choose a design” (-1). What happens next, which depends on the students’ answer, can be traced down the figure. For example, those who answer “elaborate the design” are asked to explain how they would do that and then are asked to choose between the remaining alternatives. If they answer “meet with customer” on the second pass, they receive +2 points.

TABLE I
OVERVIEW OF SIMULATION TASK

Phase	Points made	Principles
Understanding problem (see Fig. 1)	Meet with customer Prepare scenario Goal of scenario Check scenario against problem statement How and what check?	Take time to understand. Test early.
Requirements	Write requirements Review Requirements List issues in review Feasibility Tradeoffs	Test frequently. Consider feasibility and tradeoffs
Considering solutions (see Fig. 2)	First alternative response to a specific suggested solution: Consider other alternatives Compare feasibility of alternatives Compare tradeoffs. Choose one. Second alternative response to first suggestion: Think through first solution Map out components Issue of feasibility arises Consider alternatives Choose one.	Consider alternatives at first, or show flexibility if problems arise with first design. Consider feasibility and tradeoffs.
Developing Preliminary Design	Identify major components. Evaluate dependencies Refactor components to reduce dependencies Decompose major components into smaller units. Refine necessary functions.	Consider interrelations among components, reduce dependencies.
Finishing Preliminary Design	Identify problem component. Think through opportunistically or postpone, proceeding through top-down. Check against requirements and scenarios.	Opportunistic design acceptable as deviation from top-down design. Test frequency.
Developing Design	Refine components into smaller dependent units. Only additional issues should be adding functionality or refactoring to minimize coupling	Issues dealt with early do not come back to haunt.
Testing Design	Test design by building prototype. What is goal? Check that system works from user’s point of view. To check that design works.	Test frequently.
Implementatio n	Code Test components as develop. Fight with manager. Give to user for acceptance testing.	Test frequently.

Respondents are then asked to suggest to the team how they should prepare for the meeting with the customer. After they write their responses, the program presents four alternatives that the team is considering: “develop a scenario (or extended use cases or stories)” (+4), “gather information” (+2), “develop a list of questions” (0), and “write a requirements document” (-1). Respondents who do not choose the scenario option are asked to explain what they would do and then they are asked to choose among the remaining options.

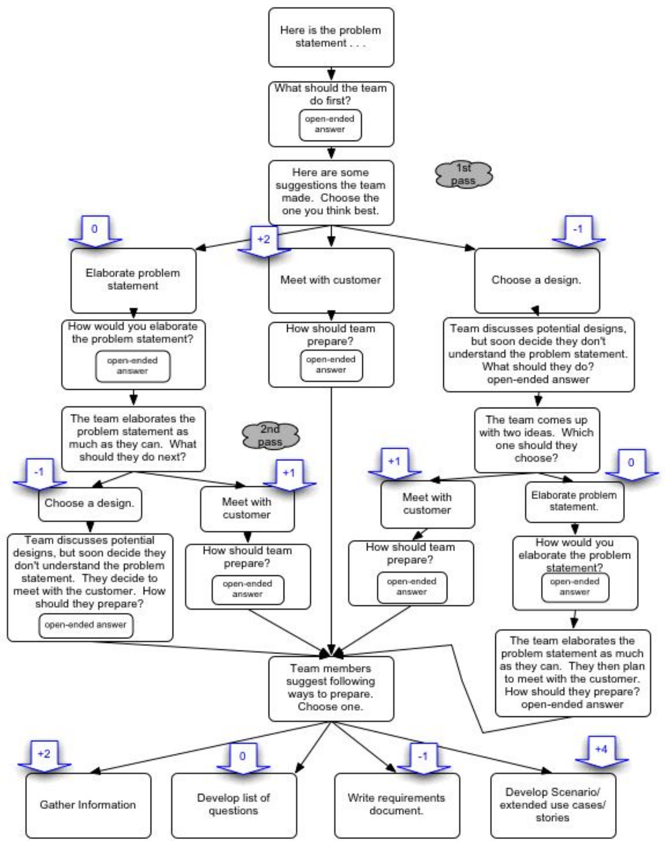

Here is the problem statement . . .
What should the team do first?
open-ended answer
Here are some suggestions the team made. Choose the one you think best.
1st pass
0
+2
-1
Elaborate problem statement
Meet with customer
Choose a design.
How would you elaborate the problem statement?
open-ended answer
How should team prepare?
open-ended answer
Team discusses potential designs, but soon decide they don't understand the problem statement. What should they do? open-ended answer
The team elaborates the problem statement as much as they can. What should they do next?
2nd pass
The team comes up with two ideas. Which one should they choose?
-1
+1
+1
0
Choose a design.
Meet with customer
Meet with customer
Elaborate problem statement.
Team discusses potential designs, but soon decide they don't understand the problem statement. They decide to meet with the customer. How should they prepare?
open-ended answer
How should team prepare?
open-ended answer
How should team prepare?
open-ended answer
How would you elaborate the problem statement?
open-ended answer
The team elaborates the problem statement as much as they can. They then plan to meet with the customer. How should they prepare? open-ended answer
Team members suggest following ways to prepare. Choose one.
+2
0
-1
+4
Gather Information
Develop list of questions
Write requirements document.
Develop Scenario/ extended use cases/ stories

FIGURE 1
STORYBOARD FOR BEGINNING OF SIMULATION TASK, WITH SCORING SUPERIMPOSED.

Respondents do not develop the scenario, but rather examine the one the team has developed. They are then asked what the team should do next. After their responses, respondents are given three alternatives and asked which action to recommend to the team: “check it against the problem statement” (+1), “check it by generating a preliminary design” (-1), and “take it to the customer” (-1). Those who choose the negative (-1) choices are presented a statement that reads “Pedro, the team leader, insists that it be checked” and are returned to the main sequence.

II. Evaluating Requirements

Respondents are not asked to develop requirements, but rather are given the requirements and asked to recommend the next step. After their responses, they are asked to choose among three alternatives, of which “review requirements” is the desired alternative. Then they are asked what issues need to be addressed and are asked to identify a feasibility issue (from four presented) and a tradeoff problem (from four alternatives). The respondent is then told that the team revises their requirements in light of the review.

III. Considering Potential Solutions

Two alternative paths are possible in this segment (see Figure 2). Respondents are told that the team has thought of a particular solution and the respondents can choose to consider additional alternatives or to proceed with the chosen one.

For respondents who choose additional alternatives, the program presents a number of potential solutions and asks respondents what to do. The program then has the team look at feasibility and tradeoff issues and choose one of the proposed solutions.

For respondents who choose to proceed with the first solution, soon a problem appears. Respondents have the choice of bulling their way through with the original, nonoptimal solution or considering alternative designs.

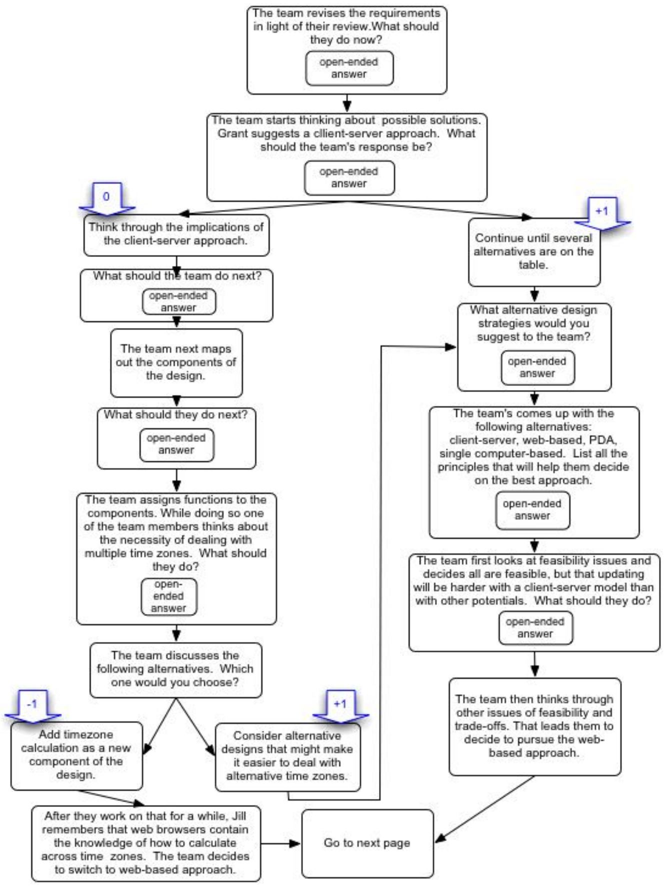

The team revises the requirements in light of their review.What should they do now?
open-ended answer
The team starts thinking about possible solutions. Grant suggests a cllient-server approach. What should the team's response be?
open-ended answer
0
Think through the implications of the client-server approach.
What should the team do next?
open-ended answer
The team next maps out the components of the design.
What should they do next?
open-ended answer
The team assigns functions to the components. While doing so one of the team members thinks about the necessity of dealing with multiple time zones. What should they do?
open-ended answer
The team discusses the following alternatives. Which one would you choose?
-1
Add timezone calculation as a new component of the design.
+1
Consider alternative designs that might make it easier to deal with alternative time zones.
After they work on that for a while, Jill remembers that web browsers contain the knowledge of how to calculate across time zones. The team decides to switch to web-based approach.
Go to next page
+1
Continue until several alternatives are on the table.
What alternative design strategies would you suggest to the team?
open-ended answer
The team's comes up with the following alternatives: client-server, web-based, PDA, single computer-based. List all the principles that will help them decide on the best approach.
open-ended answer
The team first looks at feasibility issues and decides all are feasible, but that updating will be harder with a client-server model than with other potentials. What should they do?
open-ended answer
The team then thinks through other issues of feasibility and trade-offs. That leads them to decide to pursue the web-based approach.

FIGURE 2
STORYBOARD FOR CONSIDERING ALTERNATIVE SOLUTIONS

IV. Developing the Preliminary Design

After asking respondents what should happen next, the program presents the major components of the design. Respondents' choices are: evaluating dependencies (+2), decomposing major components into smaller units (+1), and refining the necessary functions for each component (0).

Each choice leads to a different path that first asks the respondents to explain how to proceed and then asks them to choose among remaining alternatives. Depending on their choices some students see part of a sketch of decomposition with excess dependencies. Eventually they all get to the final partial sketch of the design with minimal dependencies.

V. Finishing the Preliminary Design

As the team is assigning functions, the team becomes concerned that one component will not work. Respondents have the choice of following opportunistic design or maintaining top-down design. Both answers receive 1 point and respondents are asked to justify their choice.

The program displays the preliminary design. Respondents are asked to describe the next step and then to choose between two alternatives, checking design against requirements and scenarios on the one hand, and coding on the other.

VI. Developing Design

Respondents are given four alternatives for the next step, of which the correct answer is "refine components into smaller units. The three incorrect alternatives, which are all scored –1, are "ask customer if decomposition meets their needs," "decide the preliminary design is at the right level," and "start coding."

After cycling through the remaining alternatives, respondents are asked to identify the sort of problems the team should expect to encounter at this phase. There are five alternatives, of which two are scored +1—"adding functionality to components" and "refactoring detailed design to minimize coupling while maintaining cohesion."

VII. Testing Design

Respondents are asked questions leading to choosing to develop a prototype and then are asked how it should be used. The two answers that are scored +1 are "to check that system works from user's point of view" and "to check that design works." The answer "to show a proof of concept to the customer" is scored 0 and "to serve as a first release of the product," a -1.

VIII. Implementation

The questions in this phase are designed to assess whether respondents know that it is better to test components as they are built rather than waiting until the entire produce is assembled. The narrative includes pressure from management to curtail testing.

USES IN COURSE-BASED ASSESSMENT CONTINUOUS IMPROVEMENT

The results of the simulation task can be used in three distinct ways: as assessment in the continuous improvement loop in students' skill, as assessment toward curricular improvement, and as a grading tool.

Both the multiple choice answers and the open-ended answers can be useful for continuous improvement. We recommend using the multiple choice scoring to target areas in need of improvement. Thus the instructor can identify those choice points at which more than 20% students got the wrong answer as targets for subsequent intervention (a common criterion in the continuous improvement world). Pointers toward improvement are given by the percent of students who answer the other alternatives and by the open-ended answers.

Instructors can give immediate feedback to the class as a whole, describing why the high-scored alternative is desirable at that moment. One could also inform the students of the most frequently answered incorrect feedback, although there is some evidence that focusing on errors is not the best way to build improvement [29].

The open-ended questions, we believe, are most useful for instructors to read to identify why students are making less desirable or patently incorrect choices. The answers often reveal whether students simply have no conception of the issues leading to a good choice or whether they have specific misconceptions that lead them astray.

Figure 3 shows an example of a potential problem that would serve as a basis for continuous improvement. It would be chosen as a target for intervention if more than 20% of the students failed to choose refinement as the best next step. The instructor would read the open-ended answers at three points: prior to the multiple choice, immediately after the multiple choice question, and on the second pass. The goal of the instructor's review would be to understand what the students were thinking and what might be getting in the way of their understanding.

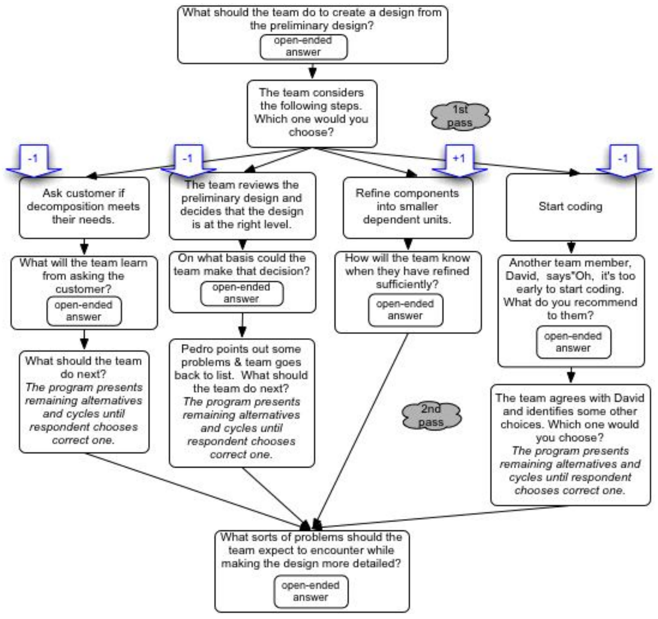

What should the team do to create a design from the preliminary design?
open-ended answer
The team considers the following steps. Which one would you choose?
1st pass
-1
-1
+1
-1
Ask customer if decomposition meets their needs.
The team reviews the preliminary design and decides that the design is at the right level.
Refine components into smaller dependent units.
Start coding
What will the team learn from asking the customer?
open-ended answer
On what basis could the team make that decision?
open-ended answer
How will the team know when they have refined sufficiently?
open-ended answer
Another team member, David, says"Oh, it's too early to start coding. What do you recommend to them?
open-ended answer
What should the team do next? The program presents remaining alternatives and cycles until respondent chooses correct one.
Pedro points out some problems & team goes back to list. What should the team do next? The program presents remaining alternatives and cycles until respondent chooses correct one.
2nd pass
The team agrees with David and identifies some other choices. Which one would you choose? The program presents remaining alternatives and cycles until respondent chooses correct one.
What sorts of problems should the team expect to encounter while making the design more detailed?
open-ended answer

FIGURE 3
STORYBOARD FOR EXAMPLE OF USING RESULTS TO FOSTER IMPROVEMENT

Instructors can also use this analysis to create their own improvement plan. For example, if more than 20% of students felt this was the appropriate time to start coding, the instructor might decide to develop activities to demonstrate to the students why coding too early has larger costs in the long run [30].

To embed this into a continuous improvement loop for students, a good plan would be to present them with a report that gives all the choice points with the correct answer and their answer and to ask them to write a reflective essay evaluating their performance and creating an improvement plan. This process could be repeated several times within a semester by developing alternative tasks for the simulation.

The simulation task was not developed as a test, but as an assessment to be used in continuous improvement. Nonetheless, it has been our experience that grading is an effective motivator, particularly in tasks in which students do not see the intrinsic value of the activity [30]. Thus, we made sure that the simulation task could be graded in a straightforward manner by summing the scores on the multiple choice questions.

DISCUSSION

This paper describes the development of a simulation task to assess students' ability to use design process knowledge in engineering design. It fills a need in engineering education for an assessment of design process skill that is suitable for course-based assessment. It assesses students' understanding of design process in the authentic context of a design task without requiring them to create the design. This has the advantage of reducing cognitive load so that the student has an unfettered chance to think about process and to demonstrate how well he thinks about it. By the same token, of course, we are making the task less authentic. It is possible that students find it more difficult to think of process when they are not also generating the design itself. That remains to be determined empirically. Nonetheless, the answer to that question is not necessarily crucial to the success of the simulation task. It might still be an appropriate pedagogical step to separate the two until students understand the design process more concretely. Once they can both suggest the appropriate activities and explain themselves in the open-ended questions, then they may be better able to apply that process knowledge while also designing. It seems certainly worth a try, because so far the evidence is that students show little design process awareness while actually developing designs [11]-[12].

This paper described use of the simulation task within course-based assessment. It could also be used as an outcome measure in research that tests various hypotheses regarding the acquisition of such expertise. For example, it could be used to study whether or how well students develop design process knowledge while engaged in design development, say in senior design courses. It can also be used as an outcome measure to gauge the effectiveness of pedagogical innovations.

Next Steps

This first simulation uses a software design task, but the structure of the simulation is designed to permit the substitution of other engineering design tasks. The overall principles—simulating a team developing a design while the student respondent recommends process activities, asking open-ended questions followed by multiple choice alternatives, and giving students second opportunities to make the correct choice—would remain the same. With different tasks, both software and hardware designs, and in different courses, instructors might choose to delete particular concepts or add others. For example, one of our participating instructors did not wish to include issues of coupling in his course, but he did wish to include a cycle of questions about searching for off-the-shelf components. Such variations are easily incorporated into the design.

ACKNOWLEDGMENT

This research was made possible by a grant from the Assessment of Student Achievement program of the National Science Foundation (DUE-0127751). The authors wish to acknowledge the help of Plamen Tchimev and Madeleine Page in developing this task.

REFERENCES

[1] Ericsson, K. A. "The acquisition of expert performance: An introduction to some of the issues". In K. A. Ericsson (Ed.) *The road to excellence: The Acquisition of Expert Performance in the Arts and Sciences, Sports, and Games* Mahway, NJ: Erlbaum,. 1996 1-50.

[2] Radcliffe, D. F., & Lee, T. Y. "Design methods used by undergraduate engineering students". *Design Studies*, 10, 1989, 199-207.

[3] Austin, S., Steele, J., Macmillan, S., Kirby, P., & Spence, R. "Mapping the conceptual design activity on interdisciplinary teams," *Journal of Design Studies*, 22, 2001, 211-232.

[4] Biswas, G., Goldman, S. R., Fisher, D., Bhuva, B. & Glewwe, G. "Assessing design activity in complex CMOS circuit design." In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.). *Cognitively diagnostic assessment*. Hillsdale, NJ: Erlbaum, 1995, 167-188.

[5] Gunther, J., & Ehrlenspiel, K. Comparing designers from practice and designers with systematic design education. *Design Studies*, 20, 1999, 439-451.

[6] Mullins, C. A., Atman, C. J., & Shuman, L. J. "Freshman engineers' performance when solving design problems," *IEEE Transactions on Education*, 42, 4, 1999, 281-287.

[7] Cross, N. "Expertise in Design: An Overview". *Design Studies*, 25, 2004, 427-441.

[8] Anderson, J. R., Reder, L. M., & Simon, H. A. "Situated learning and education." *Educational Researcher,* 25, 4, 1996, 5-11.

[9] Sims-Knight, J., Upchurch, R., Pendergrass, N., Meressi, T., & Fortier, P. "Assessing Design by Design: Progress Report 1". *Proceedings of Frontier In Education Conference*, 2003.

[10] Sims-Knight, J., Upchurch, R., Pendergrass, N., Meressi, T., Fortier, P. et al. "Using Concept Maps to Assess Design Process Knowledge". *Proceedings of Frontier In Education Conference*, 2004.

[11] Sims-Knight, J., Upchurch, and Fortier, P. "Assessing Students' Knowledge of Design Process in a Design Task". *Proceedings of Frontier In Education Conference*, 2005.

[12] Smith, R. P. & Tjandra, P. "Experimental observation of iteration in engineering design". *Research in Engineering Design*, 10, No. 2, 1998, 107-117.

[13] Sonnentag, S. "Using and Gaining Experience in Professional Software Development". In Salas, E. & Klein, G. (Eds.). *Linking Expertise and Naturalistic Decision Making*. Mahwah, NJ: Erlbaum, 2001.

[14] Batra, D., & Davis, J. G. Conceptual data modeling in database design: Similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies,* 37, 1992, 83-101.

[15] Fricke, G. "Successful individual approaches in engineering design," *Research in Engineering Design*, 8, 1996, 151-165.

[16] Gunther, J., & Ehrlenspiel, K. "Comparing designers from practice and designers with systematic design education," *Design Studies*, 20, 1999, 439-451.

[17] Fricke, G. "Successful individual approaches in engineering design," *Research in Engineering Design*, 8, 1996, 151-165.

[18] Visser, W. "More or less following a plan during design: Opportunistic deviations in specification," *International Journal of Man-Machine Studies*, 33, 1990, 247-278.

[19] Ball, L. J., & Ormerod, T. C. "Structured and opportunistic processing in design: A critical discussion," *International Journal of Human-Computer Studies*, 43, 1995, 131-151.

[20] Guindon, R. "Designing the design process: Exploiting opportunistic thoughts," *Human Computer Interaction*, 5, 1990, 305-344.

[21] Valkenbug, R. & Dorst, K. "The reflective practice of design teams," *Design Studies*, 19, 1998, 249-271.

[22] Turley, R. T., & Bieman, J. M. "Competencies of exceptional and nonexceptional software engineers," *Journal of Systems Software,* 28, 1995, 19-38.

[23] Chi, M. T. H., Bassok, M., Lewis, M., Reimann, P., & Glaser, R. "Self-explanations: How students study and use examples in learning to solve problems," *Cognitive Science*, 13, 1989, 145-182.

[24] Chi, M. T. H., de Leeuw, N., Chiu, M., & LeVancher, C. "Eliciting self-explanations improves understanding," *Cognitive Science,* 18, 1994, 439-447.

[25] Berardi-Coletta, B., Buyer, L. S., Dominowski, R. L., & Rellinger, E. R. "Metacognition and problem solving: A process-oriented approach." *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 21, 1995, 205-223.

[26] Pfahl, D., Laitenberger, O., Dorsch, J., & Ruhe, G. "An Externally Replicated Experiment for Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education". *Empirical Software Engineering*, 8, 2003, 367-395.

[27] Collofello, J., Rus, I., Ramkrishnan, V. & Dooley, K. "Using Software Simulation to Assist Software Development Organizations in Making "Good Enough Quality" Decisions". *Summer Computer Simulation Conference*, 1998 (SCSC98).

[28] Navarro, E. O., & van der Hoek, A. "Design and Evaluation of and Educational Software Process Simulation Environment and Associated Model." *Proceedings of the 18th Conference on Software Engineering Education and Training*, 2005 (CSEET05).

[29] Sims-Knight, J. E., & Upchurch, R. L. "What's Wrong with Giving Students Feedback?" *Proceedings of the American Society for Engineering Education Annual Conference,* 2001.

[30] Bernstin, L, Klappholz, D., & Kelley, C. "Eliminating Aversion to Software Process in Computer Science Students and Measuring the Results." *Proceedings of the 15th Conference on Software Engineering Education and Training*, 2002 (CSEET02.