ASSESSING DESIGN BY DESIGN: PROGRESS REPORT 1

Judith Sims-Knight¹, Richard Upchurch², Nixon Pendergrass³, Tesfay Meressi⁴, and Paul Fortier⁵

Abstract – This paper is a first progress report of a National Science Foundation Assessment of Student Achievement project designed to develop and test assessments of the ABET student learning outcome of “an ability to design a system, component or process to meet desired needs.” The overall strategy is to use the research on the nature and acquisition of expertise to guide the choice of assessment tools and then to embed those assessments in a course-based continuous improvement loop to evaluate which work best to improve those learning outcomes. So far, the project, which began in June 2002, has yielded prototypes of four assessments and a precursor to a fifth. One of those tools, a multiple-choice test of declarative understanding of design process principles, has been refined and the current version (the third of four versions) exhibits both reliability and validity.

Index Terms – Assessment, Continuous Improvement, Course-based Assessment, Design Process

INTRODUCTION

It is a truism that teachers teach to the test and students learn for the test. If our tests assess only declarative knowledge then that is what teachers will teach and students will learn. Unfortunately, acquiring declarative knowledge of course content does not guarantee that the learner can apply that knowledge. Nor are grades on design projects sufficient assessments of design skill because the quality of the product is determined by a number of factors in addition to the quality of the process.

If teachers have the ability to assess other kinds of knowledge in ways that provide pointers to improve instruction and if they are convinced of its value, they will have the tools to create a learning environment that is more likely to produce successful designers. The goal of our project is to develop assessments that support course-based continuous improvement to meet the ABET criteria of “an ability to design a system, component or process to meet desired needs.” Design in this project refers to the large-scale, ill-structured design problems typical of senior capstone design project courses and of designing in industry.

It is obvious that much declarative learning occurs during the undergraduate years. Students read textbooks, attend lectures, and demonstrate that they have learned the material on exams. They also develop cognitive maps, that is, they can display the interconnections of their knowledge by drawing complex diagrams in which concepts are nodes and connections are links and better students’ maps are more like that of experts (see [1] for a review of the use of concept maps in engineering education). Nonetheless, much research shows that acquiring declarative knowledge does not guarantee that the learner can apply that knowledge. For example, studying instructional text is not a sufficient basis for students to solve LISP programs [2]. Teaching students the principles of top-down design is not sufficient to enable them to practice top-down design [3]. Nor does assessment of declarative knowledge provide a sufficient basis from which to predict actual job performance. For example, in one review average correlations (corrected for criterion unreliability) with hands-on performance was .41 and with supervisory ratings .45 (cited in [4], p. 189).

Procedural skill, such as in designing circuits or software, has been the focus of much of the instructional research in the past few years (e.g., [2] [5]-[6]). It is clear from the existence of this research as well as from its findings that undergraduates are not attaining high levels of procedural skill, and that, when given appropriate learning environments, students can acquire such skills. For example, when starting Anderson et al.’s LISP tutors, having read the book does not help students do the first programming problem, whereas doing one programming problem improves the probability of doing a second one by 50% [2]. Even just studying an instructional example of how to construct the program produces improvements of over 60% ([7]; see also [8] for a similar finding in learning to use computers).

Although most of the research supporting the skills approach has focused on the acquisition of declarative and procedural knowledge, several disparate research endeavors have demonstrated the importance of metacognitive processing. Two main points emerge from this research. First, only highly effective students use metacognitive processing; most students, even at the university level, do not [9]-[11]. Second, simple interventions in which students are asked to process metacognitively or to explain material to oneself have been found effective in improving problem solving and, in particular, transfer to new problems [9] [12]-[13]. Pedagogically, it is not clear whether curricula need to address all these components or whether explicitly addressing some components will result in learning of all.

¹ Judith Sims-Knight, Department of Psychology, UMASS Dartmouth, No. Dartmouth, MA 02747-2300, jsimsknight@umassd.edu
² Richard Upchurch, Department of Computer and Information Science, UMASS Dartmouth, No. Dartmouth, MA 02747-2300, rupchurch@umassd.edu
³ Nixon Pendergrass, Department of Electrical & Computer Engineering, UMASS Dartmouth, No. Dartmouth, MA 02747-2300, npendergrass@umassd.edu
⁴ Tesfay Meressi, Department of Mechanical Engineering, UMASS Dartmouth, No. Dartmouth, MA 02747-2300, tmeressi@umassd.edu
⁵ Paul Fortier, Department of Electrical and Computer Engineering, UMASS Dartmouth, No. Dartmouth, MA 02747-2300, pfortier@umassd.edu
types. For example, the case-based learning movement is predicated on the proposition that procedural learning is primary and the other kinds of knowledge will develop as a result of learning procedural skills. The assessment of the success of case-based learning suggests that students are initially behind in declarative learning, but after several years, do as well as traditional students in assessments of their declarative knowledge [14]. This research has not addressed whether metacognitive knowledge is also acquired in this fashion.

An alternative point of view is that procedural knowledge builds on declarative knowledge. In this framework, the first stage of learning is the acquisition of declarative knowledge. Acquisition of procedural knowledge builds on that initial declarative base (see, for example, [15]). In support of this claim in the specific case of design, Atman and Bursic [16] found that students who read a textbook on design had better design process than students who had not read the textbook.

The goals of this project are to develop assessments of all three kinds of knowledge (declarative, procedural, metacognitive), to embed these assessments in sophomore and senior design courses, to evaluate how well they serve as (a) pointers for continuous improvement, and (b) predictors of performance in senior design courses. In other words, our challenge is to develop a set of assessment tools that can both operate at the course level to drive continuous improvement and have sufficient validity to predict actual performance at a later time. Because the senior courses will embed the same variety of assessments in their courses, we will be able to assess the predictability of each type of assessment in sophomore year to each type of assessment in the senior year (e.g., declarative to declarative, declarative to design product, metacognitive to metacognitive, etc.). This gives us the power to test a variety of hypotheses, both theoretical and practical, about the relationships and predictive power of the different components of knowledge.

THE PROPOSED ASSESSMENT TOOLS

Assessing Declarative Knowledge

We are developing two kinds of assessments of declarative knowledge—content knowledge and knowledge structure.

Content knowledge. We are developing a test of students’ declarative knowledge about the design process. Although this measure is akin to traditional tests, it has been developed according to cognitive design principles developed by Dubois and Shalin [4]. Dubois and Shalin have demonstrated that it is possible to get substantially better predictability of actual job performance if the content of the test is chosen by sampling the knowledge used by experts. Their meta-analysis of studies that predict on-the-job performance from tests found that such cognitively derived tests had average correlations (corrected for unreliability) of .72 with performance.

Knowledge structure. Experts not only know more than novices, but their knowledge is better organized. Assessments of knowledge structures take a variety of forms—rating similarity of pairs of concepts, building a graphical representation from a set of concepts, or filling in blanks of an expert's graphical representation (c.f., [17]). These can be analyzed qualitatively or submitted to several statistical techniques to quantify the relationships. Particularly promising is the Pathfinder program [18]-[19]. This technique has the advantage of being easy to administer (give the participants a list of concept pairs and have them rate the similarity of each pair) and accesses tacit knowledge, that is, knowledge that is not conscious. It is clear from the literature that much of expertise is not accessible to consciousness. As a matter of fact, that is why experts have so much difficulty teaching what they know—they are not aware of much of what needs to be taught. The resulting network diagrams can also be used to provide pointers for improvement, because, by examination of the networks of their students, instructors can identify concepts, linkages, and whole subareas where 20% or more of students are having trouble (20% is the traditional target in the quality paradigm).

Another approach to knowledge structures is to have experts and students create their own map. It has been used successfully in engineering education [1] and also provides pointers to improvement. This strategy represents the mappers’ conscious understanding rather than their tacit knowledge. It is not clear which approach—tacit or conscious—is better for assessing knowledge structure. Jean Piaget, the foremost researcher on cognitive development in the twentieth century, firmly believed that unless one's knowledge is conscious (i.e., unless one knows why one's answer is correct) it is not stable. For example, children who could not explain their answers often changed their answers when challenged or were inconsistent, giving one answer at one time and the opposite answer another time. Piaget would clearly predict that the best assessment would be one in which students construct their maps and can defend them.

In addition to the underlying issues regarding what is important in acquiring expertise, there are pedagogical and practical differences between the two approaches. The similarity rating approach is more easily administered, it can easily be scored quantitatively, and there is a fair amount of evidence that the ratings (and derived statistics) predict grades and test scores [19]. In addition it automatically produces qualitative maps that can be used pedagogically. The map construction approach consumes both more instructional time (complex maps take a while to think through) and more evaluation time. It is possible to convert a map into similarity ratings and then use the Pathfinder algorithm, but this takes time. In its favor, map construction requires students to reflect on the relationships among their constructs, an activity that has intrinsic value.

This seems an important enough issue to warrant trying both approaches and resolving the issue empirically.
Assessing Students’ Ability to Apply Process Knowledge

It is one thing to know the principles (measured by the knowledge of design test) and another to use them (cf. [13]). Furthermore, it has been found that software experts, who used excellent design process within their area of expertise (communication), were less able to apply that knowledge to a different software domain (library system) [20]. Thus, it is necessary to assess whether students can use the process even if they know the principles. Virtually the only approach that has been used is to have students engage in a design process, videotape them, and then rate the videotapes or transcribed protocols. Even if one uses a global rating scale, such as that used by Davis, Calkins, Gentili, and Trevisan [21], a minimum commitment of time would be 1-2 hours per student. This is unrealistic in a course-based assessment context.

Thus, our goal is to develop an assessment tool that will eliminate the need for this time-consuming process. We will, however, use assessment of a videotaped design task as our criterion task. From the videotape, the design notes, and the students’ final design we will develop both process and product assessment.

To create an assessment tool of process understanding that is practical within a course-based assessment context, we will develop a computerized design process simulation. This program will simulate a team designing a product. The student respondent will serve as a member of this team. The program will pause at appropriate places to ask student respondents to provide their input into the process. They will be asked both to plan the “team’s” next move and to monitor the “team’s” progress and suggest corrective action.

Assessing Metacognitive Processes

Although the role metacognitive processing plays in the acquisition of expertise has not been widely studied in educational settings, the available research (described earlier) suggests it might be a strong predictor of future design skills. It is also potentially an excellent assessment of lifelong learning in that it enables people to overcome difficulties without help, which is a mark of exceptional designers [22].

Immediately after the final design task, we will give students a self-report survey. Such scales have been found to correlate with grades and other traditional outcomes [23]-[25] and have also been used effectively in software design courses [26]-[27].

THE PLAN AND COURSE OF THE RESEARCH

Once developed, the assessments will be embedded into a continuous improvement model in sophomore and senior capstone design project courses in electrical engineering and computer science/computer engineering. The crux of this approach is to measure both process and product, analyze these data to identify problems, address those problems, and measure again. The process is repeated endlessly, because this approach assumes improvements can always be made. Both ABET and CSAB has mandated that all programs establish continuous improvement based on student outcomes.

This is the first year of this project. We have developed prototypes of the Design Process Knowledge Test, the two structure of knowledge assessments, the final design task, and a precursor of the simulation task. These tasks were given to students in electrical engineering, computer engineering and computer science classes. We are in the process of analyzing the data gathered so far. In this paper we will discuss the development of the Design Process Knowledge Test.

DESIGN PROCESS KNOWLEDGE TEST

To develop the questions of the Design Process Knowledge Test (DPKT), the coauthors first brainstormed about what knowledge of design process is and developed questions that, we hoped, would tap that knowledge. Then the test was given to available classes of engineering students. The results were subjected to an item analysis that included percent of students who got the question correct, the Discrimination Index (the percent of high scoring student s who got the question right minus the percent of low scoring student s who got it right), and the number of high and low scoring students who answered each of the false alternatives. Then poor questions were eliminated, weak ones changed to be more effective, new questions added and the revised test given to available classes. This process has been repeated four times. The data presented in this paper is from the third version (the fourth will be tested Spring, 2003).

The third version was given to 11 sophomores in a Circuits class, and three senior classes—56 juniors and seniors in three courses—software engineering, senior design course for electrical and computing engineers, and a senior computer engineering course. Students were given homework credit for completing the test, but were not graded on it. The resulting sample was 11 sophomores, 18 juniors, and 56 seniors. Of the 31 questions, 16 had DIs greater than .40 and 6 had DIs in the .30s. The Cronbach alpha for the 31 questions was .89. Thus the DPKT Version 3 (DPKT3) demonstrates considerable discriminability and internal consistency.

If the DPKT3 assesses design process knowledge that is relevant to engineering expertise, performance should increase from freshman to senior year. We gave the DPKT3 to a freshman integrated engineering class and compared the engineering majors across the four undergraduate years. Because we had far more seniors than other classes, we randomly eliminated 28 seniors. The analysis of variance of class on DKPT3 was significant, F(3,40) = 21.1, p < .001. The Tukey HSD post hoc test with alpha at .05 revealed that the freshmen were significantly worse than the upper classes.
and that the three upper classes did not differ from each other (see Table I). To assess whether the random elimination of seniors affected the results, we ran a $t$ test comparing performance of seniors who were eliminated to seniors who were not. The difference was not significant, $t(35) < 1$, $p = .82$. 

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>Descriptive Statistics Classes on DPKT3</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Mean Standard Deviation</td>
</tr>
<tr>
<td>Freshmen</td>
<td>10</td>
</tr>
<tr>
<td>Sophomores</td>
<td>11</td>
</tr>
<tr>
<td>Juniors</td>
<td>10</td>
</tr>
<tr>
<td>Seniors</td>
<td>14</td>
</tr>
</tbody>
</table>

The next step towards validity would be to see if performance on DPKT3 is related to other indicators of design process skill. Our other measures of design process—concept mapping, reflective practice, and observed process—are still being developed. In the interim we used the only design process measure available, project scores in the software engineering course. (The other two senior courses did not have a score for a major design project at this time.) The correlation between performance on the DPKT3 and software engineering project scores was significant, $r(46) = .33$, $p < .05$.

CONCLUSIONS

This is the first of a series of reports of our project, Assessing Cognitive Components of Design for Course-based Improvement. It has given an overview of the project and a report of the first assessment developed.

The Design Process Knowledge Test is a multiple-choice test of students’ declarative knowledge of the design process. It assesses the extent to which students understand the rules of “best practices” design process (such as considering alternatives before settling on one solution and evaluating the feasibility and trade-offs at the preliminary design level). The results of the third version of the test show that it has high reliability and we have two indicators of validity. First, students improve in performance between their freshman and sophomore years. Performance in the senior year is still only slightly above 50 percent, which suggests that it can be used for course-based assessment in all four years. Second, it correlates with design project scores in a senior software engineering course, thus further confirming its utility in course-based assessment for seniors. As the other assessment tools in this process become available, we will be able to answer increasingly sophisticated questions of how the knowledge tapped by this test fits into the development of design skill.

REFERENCES


