Int. J. Engng Ed. Vol. 22, No. 3, pp. 685–688, 2006
Printed in Great Britain.
0949-149X/91 $3.00+0.00

What We Have Learned in Mudd Design Workshop V: Learning and Engineering Design*

JOHN W. WESNER
Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA. E-mail: wesnerj@andrew.cmu.edu

This contribution summarises the main points discussed at the Workshop on Designing Engineering Education which took place at Harvey Mudd college 19-21 May 2005. The points raised are the meaning of the profession of engineering, the definition of design knowledge, design, inquiry and learning, learning in design and in engineering design, inspiring learning in design courses, assessing learning in design courses, programmatic and related issues in design, and apparent paradoxes. Future steps are also discussed.

Keywords: engineering profession; design knowledge; design learning assessment.

INTRODUCTION

THE WORKSHOP'S keynote speaker set the tone for the Workshop by pointing out that different disciplines actually approach learning (and doing) differently. (James W. Pellegrino, Distinguished Professor of Psychology and Education, University of Illinois at Chicago.) Engineers are *problem solvers*, which is not common to other disciplines.

As if this did not make the job of teaching engineering difficult enough, experts cannot see the world the way a novice sees the world. Thus people who are domain experts (highly educated practicing engineers, like engineering faculty members) may not be good teachers of engineering—even if they are skilled educators.

THE PROFESSION OF ENGINEERING

The Workshop began with an examination of just what is this profession (field) known as *Engineering*. The overriding theme of the presentations in this session was the roles of *analysis* and *design* in engineering. It was suggested that *design* is a specialized case of an even larger type of activity: *systems thinking*.

The attributes/roles/capabilities required from engineering graduates to meet the future needs of a wide range of potential employers were identified by an engineer practicing in industry (John McMasters, from Boeing). We are moving more and more toward needing *Deep Generalists* rather than either narrow specialists or generalists who are broad but lack depth in any area. Two studies were cited [1, 2] that identified a need to integrate the *design process* with the *knowledge* of engineering.

Who, then, will be tomorrow's engineers, and how should we prepare them? We want undergraduate education to support the development of engineering practitioners who are also good citizens. Going along with this is a belief in some quarters that engineering education should be student-centered (the student as *customer*) rather than faculty-convenient.

Despite the difficulty identifying outcomes of learning about process, data was presented that show that the identified roles and attributes are being enhanced by today's design education. These include:

- open-ended design carried out by teams
- design education
- research and analysis
- problem solving
- collaboration
- leadership
- design practice
- communication
- global perspective/awareness
- visual thinking

DEFINING DESIGN KNOWLEDGE

'System level design' has an impact on design outcomes beyond what might be expected from the fact that it involves a relatively small portion of total project effort. System-level design activities contribute the most to design success—even though they represent less than 10% of the total design effort.

We may want to broaden our understanding of *design*. Consider:

- entertainment experiences;

* Accepted 21 January 2006.

- design in fields other than engineering (e.g., fine arts);
- Software and processes.

Software Design has some real differences from other engineering design—and also some similarities. The differences may be due to the presence of more layers of abstraction in software, the enormous number of interacting parts in even a medium-sized software system, or the rapidity with which software design changes can be tried out. All engineers need to have some knowledge/skill in software design, if only for the process understanding this provides.

DESIGN, INQUIRY, AND LEARNING

There was an intriguing discussion of the similarities between learning to 'do' design and learning to play a musical instrument. While no one argued against the concept, people struggled to find specific parallels—like: What is the equivalent in learning design to 'doing scales' in learning music? Does repetition in design lead to routine, or to developing *adaptive expertise*?

Whether or not 'doing scales' (repetitive practice) helps, it does appear that learning design can be supported by:

- supporting divergent and convergent inquiry, possibly by having students develop portfolios;
- helping students to learn ways to generate ideas;
- using a variety of learning models;
- beginning to develop design concepts and skills in high school.

We learned that at TU-Delft, athletes and musicians were brought together with engineers. Words rarely heard in the engineering community were invoked: passion, drive, endurance, spirituality . . . Individuals' *level of creativity* seemingly depends upon developable factors like:

- desire and fulfillment;
- knowledge of principles;
- familiarity with objects;
- openness;
- knowledge of design and problem-solving processes.

But *can* these be developed? Not everyone agrees that design is a cognitive process that can be decomposed and taught. Some say that design is *innate*, while others maintain that since design is *non-analytical* it cannot be taught. The group consensus appeared to be on the side that says design can be taught.

The sense was also expressed that to some degree *all* engineers are designers. There may be need for a *design community advocate*, who can:

- enlarge the discussion pool;
- draw in young faculty (design and education);
- draw in more (and more diverse) potential designers;
- disseminate ideas.

What can we learn from those who are 'leaking into' and 'leaking out from' the engineering profession, including:

- practicing engineers who have not been educated as engineers;
- those who are lost to engineering after earning the BS degree;
- those who are lost to engineering before they ever get to the university?

To what extent should design education be moved beyond the engineering student body, in the spirit of developing more general *technical literacy*? More people than engineers could use the skills of open-ended problem solving.

LEARNING IN DESIGN AND IN ENGINEERING DESIGN

Design methods are tools intended for use in industry. When they are used in a course setting they must be used properly. The Design Methods that are taught should be matched to learning goals and types of projects. Despite the opposition of some 'purists' who argue for teaching only rigorous methods, heuristics are appropriate to be taught in class; we should develop students' intuition and heuristic skills.

Three tools/methods are most widely found to give positive results:

- explore different representations;
- search the space;
- iterate.

There was discussion of applying (a slightly revised version of) Bloom's taxonomy of cognitive learning to engineering design education:

- Remembering—being able to re-draw the design process.
- Understanding—being able to explain the design process and its phases.
- Applying—implementing the design process.
- Analyzing—clearly understanding each step in the process and when to use it.
- Evaluating—comparing alternate design processes.
- Creating—forming an entirely new design process.

It seems to be already pretty well accepted that design projects should drive the students to consider not just the design itself, but also skills like Project Management, Teamwork, Communications, Design Processes, Research, Analysis, Evaluation, and Decision Making. All together, this is a very full palate. How can all of this be achieved? One approach is to treat the creation of a design course as a design project in itself, and to employ all of the familiar design tools. This ought not only to lead to a better class; it should also help design faculty to become familiar with the tools they will be teaching.

Even within the context of the design project itself, we may want to encourage building, adapting and exploring.

The question was asked, what about development of *Systems Engineers*? Is there a role in the formal education process, or does this happen in practice?

INSPIRING LEARNING IN DESIGN COURSES

If we accept that design can be taught, how can we help to achieve that goal? Some suggestions are:

- emphasize the 'design' content in all sorts of activities, including education research;
- emphasize the 'human element' in design—e.g., assistive technology for the aged or infirm—as a motivator;
- custom design 'needs driven' learning experiences.

One characteristic that seems to work in design classes is *not* to employ the common classroom mode of 'giving students information'. It seems better to get them doing things. This reminded the author of a commitment made at an earlier Mudd Workshop, 'never to lecture again'.

Continuity and student confidence may be important issues. Many programs now have both ABET-mandated 'Capstone' design project courses and 'cornerstone' (first year) design courses, often involving the students in simple projects. There seems to be a need for connecting courses (middle pieces) in between these two.

ASSESSING LEARNING IN DESIGN COURSES

One reason for this different approach is that what motivates students in design courses is often different from what the faculty want to convey and assess. It also seems true that student performance in design courses does not necessarily indicate the extent of their learning. They may apply to their design 'project' work only the bits of learning they sense are appropriate, and miss the advantages that might accrue from applying even more. A good example is the extent to which students do or do not apply the analysis tools they have learned to their developing designs.

This leads to asking whether our assessment methods could be used by the students to assess their own learning? Would this enhance learning, especially autonomous learning?

The question was also asked, if there could be a common basis for assessment that might be applicable to a variety of design courses. This led to the concept of developing a repository for design education rubrics, assessment, experimental tasks, etc. One participant volunteered to begin such a repository with information other Mudd V participants would send to her.

PROGRAMMATIC AND RELATED ISSUES IN DESIGN

Some practitioners view participation on design teams to be a validation of their roles as *specialists*. If these specialists remain isolated and do not share their special knowledge with other team members, they may help the project deliverable but not the professional growth of the entire team. To what extent does this matter? To the extent that it does, how can we convey the most appropriate behavior in student design experiences?

As mentioned earlier, it seems already to be accepted that design projects should drive the students to consider not just the design itself, but also the *soft skills* (e.g., Project Management and Communications). Professional Growth does not seem to be a big stretch beyond this.

There was discussion of the need for *technology literacy* by the entire populace, and how this might promote some sort of culture change.

Apparent paradoxes

An interesting issue that emerged during the *wrap up* discussion was that during the course of the Workshop several apparent paradoxes emerged:

- the tension between *rigor/process* and *heuristics;*
- the need to encourage both *divergent* and *convergent* thinking;
- the need to be simultaneously *systematic* and *creative;*
- the need to perform both *analysis* and *synthesis;*
- the need for the design engineer to be both a *generalist* (a 'deep' generalist!) and a *specialist;*
- the need for the design engineer to take both a *systems view* and a *detail view;*
- the need to balance *passion* and *money;*
- the need to promote both *competition* and *collaboration.*

NEXT STEPS

In the spirit of having this Workshop serve as more than a pleasant discussion forum, the Workshop participants were given one final assignment. They were divided into several teams, each of which was asked to:

- brainstorm what they believed to be the key outcomes of the Workshop;
- using the tool *Affinity Diagramming*, identify the higher-level outcomes indicated by the results of their brainstorming;
- develop (over lunch) individual or multi-person commitments for action on their high-level outcomes.

Some of the 'high level outcomes' that they identified are:

- improve our definitions of *design;*
- find 'best practices';
- the importance of 'systems thinking';
- How do people imagine a profession of design?
- Why do people enter and leave engineering?
- consider the societal implications of engineering design (engineers as 'society's technical problem solvers');
- be aware of the 'emotional layers' of design (passion, spirituality);
- better define the attributes of *design education;*
- consider the student as a 'customer';
- investigate the pedagogical implications;
- nurture our own design community ('build a big tent');
- implement changes in curricula and in courses;
- include more design in undergraduate engineering education;
- help students to take responsibility for their own learning;
- learn more about the tools, and share them;
- share assessment methods;
- get some conversations going on the issues we discussed.

When the assembled participants were probed at the end of lunch, there were fewer commitments made than at previous Mudd Workshops, and none at all that could be termed 'daring'. Most focused on collecting ideas from the Workshop participants, and sharing these around.

REFERENCES

1. *Pedagogies of the Professions: Engineering Education in the US,* The Carnegie Foundation for the Advancement of Teaching.
2. L. R. Lattuca, et al., *Are the EC2000 Curriculum Changes Consistent with What Employers Consider to Be Important?* Data from the National Study of EC2000.

John W. Wesner teaches Mechanical Engineering at Carnegie Mellon University, where he has been since the fall of 2000. Wesner received his BS and his Ph.D. from Carnegie's MechE Dept., and an MS in MechE from Caltech. Retired from Lucent Technologies Bell Laboratories, Wesner's interests are in many aspects of Engineering Design, especially Product Development and improvement of Product Development Processes. Teaching satisfies Wesner's long-term desire to share his engineering experiences. Currently, Wesner also is ASME Vice President–Nominee for Programs and Activities in the newly-formed 'Knowledge & Community Sector'. During 1996–99 he served as ASME Vice President for Systems and Design. Outside of his professional activities, Wesner is a passionate (and published) model railroader.