

## *THE ULTIMATE GUEST SPEAKER: A MODEL FOR EDUCATOR/PRACTITIONER COLLABORATION*

Josh Tenenberg  
Computing and Software Systems  
Institute of Technology  
University of Washington, Tacoma  
Tacoma, WA 98402  
(253) 692-4521  
jtenenbg@u.washington.edu

### **ABSTRACT**

This paper describes a unique model for bringing professional practice into the computer science classroom. It does so by pairing a practicing professional with a practicing educator to plan and teach a course, thereby exploiting what each does best. This model was instantiated in 2009 between the author and an interaction designer from Google.

*Presented at the Eleventh Annual Northwestern Regional Conference of the Consortium for Computing Sciences in Colleges, Tacoma, WA USA, October 2009.*

### **TEACHING SOFTWARE DEVELOPMENT AS TWO PRACTICES**

Short shameless confession: with each passing year, I get further from the practice of software development, even as I get deeper into the practices of teaching and research. In this paper, I describe a pilot project to address the issue of bringing professional practice in software development into my classroom while at the same time leveraging my expertise as an educator. The idea is simple: pair a practicing professional with a practicing educator to plan and teach a course. Exploit what they each do best. I discuss this idea in two ways. First as a personal narrative that describes the background and motivation, and second as an experience report of a pilot version that I carried out in teaching a human-computer interaction design course with an interaction designer from Google.

### **PERSONAL NARRATIVE: PRACTITIONER EXPERTISE**

The idea began five years ago, when an ex-student, Adam Barker, moved to Seattle to begin work with Amazon. It began, in fact, shortly after he entered my software engineering classroom to run a session on paper prototyping. At that time, I had taught computing for a decade and a half. My graduate studies were in the 1980's, a rigorous program in computer science, specializing in good-old-fashioned-AI (GOFAI). This was preceded by a brief (less than one year) term as a professional software developer, writing contract code in Basic (of the non-visual kind) on PC's running CPM. During the 1990's, I taught core computer science courses in a computer science department that I would characterize as "traditional" in its approach, in a regional campus of a midwestern university. I was relatively unconcerned that the overwhelming majority of my students would be professional software developers for a living, and that they would survive in that profession largely as a result of learning a number of other practices outside of the

formal setting (e.g., in analysis and design, in interacting with users and clients, in navigating a hierarchical organizational structure, in written and verbal communication).

My perspective began to shift when I moved to my current university in 2000. The degree program had a more applied focus, and I found myself teaching such things as software engineering, database design, and human-computer interaction. I immersed myself in an academic literature with its focus on process, reducing uncertainty, measurement, and standardization. But I equally immersed in a practitioner literature discussing “craft” knowledge in such areas as heuristic guidelines for object-oriented class design, the use of pattern languages, and how-to guides for such things as user stories, unit testing, and refactoring.

What I began to suspect was that, important as computer science is to software development, the relationship is as mathematics to civil engineering: essential, but insufficient. Developing depth in teaching and research in computer science made it difficult for me to also develop the deep, craft knowledge that comes from doing software development full-time, day after day, within a community of other developers under real-world constraints. By reading the literature, I had developed what Collins [1] calls *interactional expertise* i.e., I could talk the talk. But by not actually having several years of professional practice (and the professional practice that I did have was 30 years ago), I did not have what Collins calls *contributory expertise*, i.e., I could not walk the walk.

This problem is most acutely felt, I suspect, by university teachers in disciplines closely linked to the professions: engineering, law, medicine, social work, nursing, management. Our primary practices at the university level are in teaching and research, not engineering, law, medicine, social work, nursing, or management. Many such teachers maintain an active practice in the discipline. And some from studio-based professions teach in an *atelier*, a studio where the master and apprentice work together on real-world projects, offering opportunities for continuous critique and interchange between master and apprentice. Yet many, like myself, invested their effort instead in research within a subfield and increasing their skills in teaching; the demands of tenure and promotion made it difficult to do otherwise. There was simply no more of me to also develop as a practitioner in the field.

Adam, on the other hand, having graduated in the mid-90's, had spent several years cutting his teeth on challenging projects in a variety of software development organizations. He had moved far beyond the software development skills he had mastered as a computer science undergraduate at Indiana University South Bend, where we had first met. He could most emphatically walk the walk; he was, in fact, no longer walking, but running at a rapid clip. His interests had shifted from the direct writing of software to HCI design, and he was moving up the professional hierarchy in terms of the responsibilities that he was being charged with. So, in early 2003, when he moved to Seattle, I invited Adam to come to one of my class sessions. He had, a few months earlier, encouraged me to read the book *Paper Prototyping* [5], written by a practitioner in interface design. I asked Adam to run a session where students would develop a paper prototype for the system that they were currently designing in my software engineering class, and Adam would demonstrate how to run usability tests of their paper prototypes.

What happened in that classroom caused a permanent shift in my perspective. I was confronted both by the power of the expertise that Adam possessed, and faced with the expertise that I lacked. His expertise was not expressed discursively in a brilliantly

delivered lecture, which is often what we expect when we invite guest speakers. Neither was it expressed as a set of abstract principles. Rather, it came out when he and the students discussed student-produced designs directly in front of them. For in responding to the student work, Adam was engaged in the very materials of his practice. He said things like “you have a lot of blank space here; interface designers talk about ‘the tyranny of the void’”, and “don't start with that splash screen; take your user right to the information that you want them to see.” The interaction between Adam, the students, and the materials at hand—and the vicarious learning for the rest of us—was reminiscent of those between masters and apprentices in the design disciplines described by Schön in *Educating the Reflective Practitioner* [6].

How else do we learn this craft knowledge? It is deep and subtle and derives from immersion within a practice community [2]. Adam saw things in their interface designs that I simply could not see. And he could guide them in stepping through a usability study—what the facilitator does, what the facilitator says, how the facilitator ensures that users keep talking—in ways that I could not do. Certainly I knew the steps, and I could work my way through the dance. I could even lead the students in this dance, inelegant though it might be. But I could display no artistry, none of what Schön calls *knowing-in-action*: “I shall use knowing-in-action to refer to the sorts of know-how we reveal in our intelligent action ... We reveal it by our spontaneous, skillful execution of the performance; and we are characteristically unable to make it verbally explicit.”

When, five years later in 2008, I invited Adam back into the classroom to comment on final projects of students in my Human-Computer Interaction course, I experienced even more strongly his professional artistry. For in the span between his two classroom visits, he had moved to Google in the Bay Area, spent several years doing interaction design there, working on a number of projects including as lead designer of Gmail chat. He had also begun to think about the kinds of learning and teaching activities that he might construct within his professional setting in order to educate his engineering-centered colleagues about the key practices of interaction design.

And thus an idea began to emerge about how to combine our expertise in the university classroom.

## **THE IDEA**

The idea is simple: a practicing university faculty member and practicing professional jointly plan and carry out the teaching of a course related to the professional's domain of expertise. The faculty member retains full responsibility for all academic aspects of the course: planning the syllabus, developing the assignments and examinations, and assigning grades. The professional joins the faculty member in the classroom on a regular basis (though not necessarily every class session) and evaluates a sample of the student work on an advisory basis. Rather than bringing the students to the master in the atelier, it was bringing the master to the students in the classroom.

## **INSTANTIATING THE IDEA**

The idea was formed in winter 2008, shortly after Adam's visit, and Adam indicated an interest in participating during the following academic year. During spring 2008, I developed a proposal for the appointing of *Industry Fellows* in the Institute, which is the name we give to the professional who takes on the role specified in “The

idea” section above. With enthusiastic support from my departmental colleagues, I sought input from our Institute Advisory Board (consisting of education, government, and technology industry leaders in the Puget Sound region), who also endorsed the idea. My department chair also strongly supported the idea, and provided funding so that Adam could be paid at approximately half the rate at which we pay part-time instructors to teach a course. This was far less than the market rate for his skills, but sufficient to symbolize institutional commitment. The reasons for having the lead time of one year between the initial planning and the actual running of the course was first, because university time-tabling happens months in advance; we needed to ensure that the course would be scheduled during the term and during the times to accommodate Adam's schedule. And second, because we needed to protect the class session times in Adam's schedule, which is often planned far into the future.

During the summer of 2008, Adam and I met for three extended meetings, where we planned the course and the coursework, a human-computer interaction design course in the Institute of Technology at the University of Washington, Tacoma scheduled for the following winter. Although we preserved much of the overall structure and most of the topics of the course as my earlier offerings, we significantly reworked the focus and the primary projects that the students were to undertake. By the end of summer, we had a completed syllabus as well as all course materials. In addition, we had a list of work products that Adam would be bringing in to demonstrate to students developed from his own practice: personas, field notes from usability studies, wireframes. In addition, he prepared a longer case study concerning the design decisions that he and his team made in developing the Gmail chat interface (elaborating the presentation of the same case from our associated class reading from Chapter 6 of [4]).

I designed the course with two key constraints in mind: that it be sustainable and that there be a clear division of labor. The design needed to be sustainable in the sense that the demands on Adam would be relatively small (a few hours per week) and evenly distributed throughout the term. We met this by determining that he would attend only one of the two class sessions each week. I also meant for the model itself to be sustainable, in that it required only incremental resource and effort by *both* Adam and myself, increasing its likelihood of being adopted by others and transferred to different settings. The division of labor was along lines of expertise as well as institutional authority. I maintained formal control over the academic decisions, with Adam serving in an advisory role. This was made clear to my colleagues, administrators, and students. As importantly, and consistent with my interests in sustainability, I performed all of the activities associated with my expertise as an educator at my institution. These included such things as writing and grading the assignments, writing the syllabus and integrating this with institutional requirements, and getting copyright clearance on the reading pack. Adam's time was therefore freed up to exploit his expertise: discussing cases brought from real projects that he had worked on, running a mock usability test in the classroom, talking through field notes, wireframes, and personas that he had developed on past projects, and critiquing student work.

The running of the course itself had the following pattern. During the first class session of the week, when Adam was not attending, students and I would discuss the course readings (generally practice-based), do in-class exercises related to the skills that they would be exercising on their team-based group project (e.g., interviewing,

observing, sketching, brainstorming, prototyping), and clarify any questions related to the course assignments. During the second class session of the week, when Adam attended, there was usually a presentation of an artifact drawn from Adam's work practice, as well as a critique of each student group's project to date. In this regard, the sessions that Adam attended took on some of the character of a studio-based approach [3], with its public critique sessions and the learning interactions between master and apprentice mediated by the materials of design. Between the two class sessions, we talked on the phone for 30 to 60 minutes, discussing student progress, what we would be doing during the following class session, and our separate responsibilities in preparing for and leading the different parts of the session. We also would briefly discuss the class session after each one Adam attended, and would have a longer debrief after each of the three student milestones (roughly every three weeks).

## RESULTS

I am writing this paper during the last several weeks of the term in which it is being run. The “results” are thus tentative, and, like the rest of this paper, told from my perspective as a faculty member. I neither have the data nor consider it in scope to report on the impact of Adam's participation for Adam or for the students, though I hope to do so in the future.

Adam was able to meet each week with the class, save for one week, and he attended both sessions during the final week. Student attendance was never less than 90% in any class session.

I experienced the benefits of working together even during the planning phase. Being the only person in my department who teaches this course, I was happy to have a colleague with deep knowledge of the area with whom to discuss my key design decisions and their rationale: the sequence of topics, the readings, the assignments, the instructor-led demonstrations and student practice sessions. Having a working professional critique the course helped me to make explicit my tacit assumptions about what I wanted to happen. It also helped to legitimize many of the aspects of the course that I had already been doing and gave me greater confidence in the design that we developed<sup>1</sup>.

Teaching the course was a joy. This is partly due to simply having someone else co-teaching with me (the tag-team effect). Having worked out the division of labor in advance also reduced possible causes of conflict. I was fortunate, in that Adam was clearly committed to carrying out his part of the bargain, even when things were hectic at work, including the launch of a major new software product during the term. I was fortunate as well that he had an engaging interactional style.

What was most rewarding, however, was the amount that I learned about the subject matter by watching Adam's interaction with the students and the material. I would write down his verbatim comments that he said to the students at each phase in the design project: “scope it smaller” as they were doing user inquiry, “why did you make that choice instead of this” when they were prototyping, and pithy rules of thumb when they were building specific interface elements (“don't use a modal interaction there, since it removes the user from the surrounding context”). And I think (though am not certain), that Adam as well learned about teaching at the university, that it involves complex sets

---

<sup>1</sup> All course materials can be found at: <http://faculty.washington.edu/jtenenbg/courses/452/w09/>.

of design choices under a number of constraints. Ultimately, I believe that we provided simultaneous and mutual socialization into our respective communities of practice.

### **EXPERTISE, AGAIN**

If I display professional artistry in any setting (and on my good days I think that I do), it is in educational practice. The knowledge about teaching and learning that I have acquired is both of the general kind—about cause and effect, about the way minds work, about different methods of presentation and assessment—and of the specific kind. This specific knowledge includes such things as the characteristics about the classrooms in which I teach and the resources available there, how much paid work my students do outside of class and what kinds, their educational background, the amount of work that I can assign from week to week so as to push them along yet not be met with outright rebellion, the scrutiny that my choices will receive from my colleagues, to name just a few. And this combination of general and specific knowledge, if it is to actually be *expertise* (let alone artistry) must be actively brought to bear on the practical problems of what I ask students to do, what occurs in my classroom, and how I assess student performance.

Adam's expertise is in interaction design (as it previously was in writing code), drawing upon general and specific knowledge to enable him to practice skillfully within his professional setting. Although he can read about teaching to develop interactional expertise, without actually teaching, he will not develop contributory expertise, deep knowledge about teaching embodied in activity situated in a particular time and place, which is what we would hope instructors to have when we entrust them with classrooms of students.

But working together, with clear areas of separate responsibility as well as overlap, we leveraged our individual expertise into something that was much better (from my vantage point) than what either of us would be able to do alone.

### **ACKNOWLEDGEMENTS**

Thanks to Janet Ash for the title, to Sally Fincher for reading and commenting on an earlier draft, to Adam Barker, for his enduring commitment to education, his insight, and professionalism, and to the students in the HCI course.

### **REFERENCES**

- [1] Collins, H., Evans, R. *Rethinking Expertise*, Chicago, IL: University of Chicago Press, 2007.
- [2] Lave, J., Wenger, E., *Situated Learning*, Cambridge, UK: Cambridge University Press, 1991.
- [3] Reimer, Y.J., Douglas, S.A. Teaching HCI design with the studio approach. *Computer Science Education* 13, 191-205, 2003.
- [4] Saffer, D. *Designing for Interaction*, Berkeley, CA: New Riders, 2007.
- [5] Snyder, C. *Paper Prototyping*, San Francisco, CA: Morgan Kaufmann, 2003.
- [6] Schön, D., *Educating The Reflective Practitioner*, San Francisco, CA: Jossey-Bass Publishers, 1987.