

“It’s Just a Method!”

A Pedagogical Experiment In Interdisciplinary Design

Steven Harrison

Dept. of Computer Science,
Virginia Tech
510 McBryde Hall - 0106
Blacksburg, Virginia 24061
(540) 231 7783
srh@cs.vt.edu

Maribeth Back

Fuji Xerox Palo Alto
Laboratory
Palo Alto, CA 94304
(415) 342-2832
back@fxpal.com

Deborah Tatar

Dept. of Computer
Science, Virginia Tech
508 McBryde Hall - 0106
Blacksburg, Virginia 24061
(540) 231 8457
dtatar@cs.vt.edu

ABSTRACT

What does a student need to know to be a designer? Beyond a list of separate skills, what mindset does a student need to develop for designerly action now and into the future? In the excitement of the cognitive revolution, Simon proposed a way of thinking about design that promised to make it more manageable and cognitive: to think of design as a planning problem [11, 29]. Yet, as Suchman argued long ago [32], planning accounts may be applied to problems that are not at base accomplished by planning, to the detriment of design vision. This paper reports on a pedagogy that takes Suchman’s criticism to heart and avoids dressing up design methods as more systematic and predictive than they in fact are. The idea is to teach design through exposure to not just one, but rather, many methods--that is, sets of rules or behaviors that produce artifacts for further reflection and development. By introducing a large number of design methods, decoupled from theories, models or frameworks, we teach (a) important cross-methodological regularities in competence as a designer, (b) that the practice of design can itself be designed and (c) that method choice affects design outcomes. This provides a rich and productive notion of design particularly necessary for the world of pervasive and ubiquitous computing.

Author Keywords

Design methods, design critique, design review, design education, design realization.

ACM Classification Keywords

H5.2, H.5.m. User Interfaces: User centered design, interaction styles, input devices and strategies, design methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DIS 2006, June 26 – 28, Penn State University.

Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

INTRODUCTION

A major pull in the field of design over the past thirty years, stemming from Simon’s notion of a design science [29], is to encourage designers to see themselves as involved in a controlled, manageable and cognitive enterprise. Consistent with this is the approach of many HCI design textbooks and syllabi cultivating a particular paradigm, especially those featuring engineering and social science skills. For example, Rosson and Carroll [26] use scenario-based design to develop the capacity of their students as observers and analysts. Yet, recent books from Paul Dourish [10] and Malcolm McCullough [19] point to changing paradigms of human computer interaction that particularly emphasize the importance of design factors “outside the box”. Arguably, to truly be an HCI designer means to be able to see the differences between approaches, their underlying principles, and make use of their methods appropriately.

In prevalent pedagogies, students enact a method specific to the paradigm (in a project or small exercises), then use that experience to explain the underlying idea of the paradigm. This idea is straightforward but has a significant downside; students encounter a “cookbook”, a sequence of (what they see as) immutable, predetermined steps as their first exposure to human values and behavior.

We address this downside by teaching many contrasting methods. Each is a set of rules or behaviors that produce artifacts for reflection and development. Each provides a lens through which the members of a design team build an understanding of activity constituents.

Where a single method limits allowable rhetoric, values, and results, shuts down thinking about process, leaves the impression of being a complete explanation, is particularly problematic for the inexperienced, and therefore sometimes produces unfortunate results, multiple methods do none of these. Multiple methods emphasize the context of effectiveness in which particular methods function. A plurality of methods is one means to understand that designing is composed of different activities and that these activities have names, and using the names of the activities enables

collaboration. Contrast and reflection are key components to method use.

An Analogy

We can illustrate the merits of multiple methods in the practice of design by an analogy. Two main approaches to teaching reading to children are whole word and phonics. Entrenched positions debate their merits. Each is both a theory and a method; each is, at a theoretical level, incompatible with the other. However, arguably, a good teacher will start with one, and if it does not work, switch to or include elements of the other. Such methodological switching is often politically charged, but highly pragmatic.

Human-computer interaction is awash in methods and the theories that underlie them. Like the phonics/whole word debate, they are often at philosophical odds with each other with entrenched camps. But if we are to train designers to think across disciplinary boundaries and be facile with the activities of design, we must get them to see all the perspectives on designing. We must teach them to organize activities into the malleable units we call methods. But we must also remember that the purpose of design is not the method itself, but the product. Just as reading transcends the pedagogy used in primary school, design must transcend the methods used to learn it. Methods are useful in that they give activities names and create a reflective framework for design work.

“It’s Just A Method” Method

The “it’s just a method” method is about the boundaries, utility, appropriation, and negotiation of processes that comes from actively performing many different methods over the course of a project. Therefore, the pedagogical approach we take is to provide a tidal wave of methods of many sorts, to learn the methods without commitment to their underlying principles, and to drive reflection following their use. Using project-based learning¹, this HCI pedagogy focuses on the *idea* of methods as a means to understand what is and what is not under the control of designers.

To teach students to discriminate between methods --- and to create dialogue, to legitimize the value of anticipating and planning design process, and to develop some shared understanding and common ground --- critique must be incorporated as part of the design process. This critique is in terms of the results of the method. It is only then that process reflection is appropriate – asking “What did this method do for our project? What can we do differently next time?”

Methods in HCI Design and Design Education

Although many methods are usually considered central to HCI design, and some advocates strongly argue for particular methods, there is little research on the value of methods

to the practice or learning of designing. In fact, there is little research on this in any design discipline.

Two important works examine design in general and suggest the utility and role of meta-methodology in practice: John Chris Jones' *Design Methods* [16] (particularly the second edition with its reflective and extensive introductions) and Don Schon's *Reflective Practitioner* [27].

Jones -- through use of non-reductionist rhetorical devices such as parable and the random rearrangement of sentences and paragraphs, and through a deeply “engineered” structured encyclopedia of methods -- presents a case that designers have the means to design their own process of design.

Don Schon makes a strong case for the use of explicit post-facto reflection. This is both a method and a meta-method since it incorporates the ideas of reflection upon the act of method and the utility of reflection.

Additionally, some methods implicitly or explicitly involve tailoring and reflection. For example, participatory design (PD) calls upon the designer to find the appropriate method with which to listen to the user [7]. Bjerknes and Ehn have written about the reformulation of PD methods for specific partner-users in specific contexts.

While the meta-method of PD establishes an entire design approach, some domains have developed methods for particular aspects of designing. A classic engineering example is Pahl and Beitz' method of concept evaluation [24]. Closer to home, design rationale [22] uses the idea of understanding design history in order to better design designing (and design better artifacts, of course). The atoms of design rationale however are often orthogonal to the method by which they arise. For example, rationale might capture the use of a morphological box methodology and its outcome, but not the criteria by which the use of the method in that circumstance occurred or how the aspects of the design space were arrived at.

Fallman [12] also has written about gaining control over one's designing practices. Interestingly, Fallman uses sketching to deconstruct designing and place it in a third “not-art, not-science” category of creative action. This brings us back to the overall idea of the program and the motivation for the course – placing interaction design into a diverse designerly setting.

Lastly, most of the methods discussed by HCI researchers and traditionally taught in schools focus on data-gathering and developing requirements. For example, DIS '04 devoted two sessions (four of the six papers) to methods in this area [8, 20, 28, 30]. This ghetto-izes the role of methods, leading to the assumption that the topic of methods is restricted to only requirements-gathering and the rest is just some activity that all carry out in exactly the same fashion.

APPLYING THE “ITS JUST A METHOD” METHOD

We have applied this method in teaching project-based interaction design courses in the computer science depart-

¹ See [35] for a good overview of the wide-spread status of project-based learning (“PBL”) in engineering.

ments at UC Berkeley and Virginia Polytechnic Institute and State University (“Virginia Tech”). While two were graduate and one was an undergraduate class, in all cases, an introduction to HCI was a strongly-recommended prerequisite. At Berkeley, the prerequisite class had taught user-centered design, while scenario-based design was taught at Virginia Tech.

Class 1

The course called “Design Realization”, at UC Berkeley in the Berkeley Institute of Design [1], was designed to prototype the second core class that master’s and Ph.D. students would take upon entering the BID program.

BID founders recruited existing graduate students with the idea that this would introduce them to ideas that they would not otherwise encounter. Since many were established doctoral students, BID founders also felt that encompassing some of the content from the missing preceding class would not create an overly burdensome workload.

These problems were an advantage for the issues the study reports here since it meant that the students came in with deeper and narrower expertise to be “overcome” and the bits of representation and methods to be taught were not misrepresented as ends in themselves.

Students’ Backgrounds

Fourteen students took the course. Seven were in Computer Science, five were from an informatics program that shares some classes with the HCI program, one was an engineering exchange student from a Scandinavian technical university, and another was from Art Practice. A few knew each other previously.

Since the class was not required and was a bit of an outlier from the core program in human-computer interaction, it was informally represented to prospective students as a concept design class for interactive products. Therefore, many of the students who took the class self-selected based upon some long-term career interest and/or previous experience. For example, one informatics student was an accomplished graphic designer and a couple of computer science students were concurrently taking a computer science class in cartoon-style animation. Almost all had great programming skill, some having had professional programming experience.

Coursework and Team Projects

The official syllabus describes a class in which student teams use an iterative process to design and create working prototypes of interactive devices. The form, setting, users, purchasers, design problem, design methodology, and technologies are not specified.

To create project teams, students were surveyed and placed in one of four categories based upon primary skill-set, interests, and “presentation”². Teams were self-organized,

but had to have participants from each of the four categories. Since at the time of the selection process there were 16 students, four teams of four each were expected. (Two students later dropped out, leaving three teams of four and one of two members.)

While the team formation method was intended to create more-or-less balanced skill set teams with some diversity in disciplinary knowledge, the actual formation centered on proposals for projects.

The Project

The term project was to design something that embodies the idea of “instrument”, where instrument combines the qualities of musical and scientific instruments (playable, but using datasets rather than musical sounds). Instead of a brief or problem, the project asked for an investigation of the nature of “instruments” and to find a large database against which the instrument could be used or played. The entirety of the device or system had to be realized with elements that meshed in appearance and behavior with the overall idea. While no particular interaction technique or technology was specified, standard input devices like keyboards and mice were not permitted.

Class Flow

The class met twice weekly for 90 minutes. There were a few short homework projects done individually, mostly clustered in the first half of the semester. The team projects were developed outside of class hours; however, three official reviews were placed on the course calendar. The first few course meetings were devoted to reviewing available technologies and filling in some of the missing material from the non-existent precursor course. Other than reviews of the team projects and discussion around the homework, the rest of class time was spent discussing a variety of design methodologies, their strengths and weaknesses, and experimenting with trying out a few of them in the course of the project design.

Following the pattern set in many of the computer science and HCI classes they had previously taken, the students initially framed the class project as an optimization problem for which they only needed to draw upon narrow, deep knowledge of systems and programming to create an efficient implementation.

Class 2

The second was an undergraduate course in the Design of Information at Virginia Tech. Its main focus was on the architecture of information and the ways in which information can be presented and understood. Therefore, its course goals included relating the construction of meaning to practical implementation issues.

² “Presentation” was a catch-all subjective analysis of class participation to date, amount of obvious thoughtfulness in class discussions, etc.;

for example, vocal protestation of a particular design exercise rated “high” as was any demonstration of knowledge.

Students' Backgrounds

Like the graduate students at Berkeley, the students came into the class with strong backgrounds in software engineering. About 30 students enrolled in the class, mostly from computer science; a few were human-factors majors and two were communications majors. Some knew each other previously. Almost all had great programming skill, some having had professional programming experience.

Very few students brought to the class much skill or experience in creative presentation of information. However, a few students were facile with Flash animation and web-site design tools.

The Project

The term project was to design an *alternative* website for each of the departments in the College of Engineering. The goal was to better inform freshmen in the College when selecting a major. "Alternative" meant one that was in parallel with but did not replace the official departmental website; it had to speak to freshmen. Unlike the project in class 1, this would use familiar genres and interaction techniques, but it still required students to deal with design issues outside of a traditional problem-solving mode. Given the audience and goal, each was to have elements of narrative as well as structured information.

Coursework and Team Projects

Students were organized into ten project teams of three students each. Each team was randomly assigned a different department to research and create a website for.

A methodology similar to the one used in class 1 was used to create project teams: students were surveyed and placed in one of three categories based upon primary skill-set, interests, available resources, and "presentation". Teams were self-organized, but had to have participants from each of the three categories.

In most homework projects and incremental reports, students could earn no better than a "B" if they met the specification; to get an "A", they would need to come up with some unexpected addition. Many quickly learned that employing a design method that was not specified in the assignment would earn the "A".

Class Flow

The class met three times weekly for one hour. Unlike the previous class, methods were much more explicitly part of the schedule. Most Friday classes were devoted to exercises that made use of one or another method. In the first half of the semester, individuals created and used a database of images of information in the everyday environment. The other two class meetings each week were fairly traditional lectures and discussion on information architecture and scaffolding the development of the project.

The team projects were developed outside of class hours; however, four official reviews were placed on the course calendar.

Class 3

The third was a graduate course in the Design of Interactive Systems at Virginia Tech. It was patterned on class 1 so it focused on realizing a system that involved interaction, avoiding the window/mouse/keyboard paradigm. Malcolm McCullough's "Digital Ground" [19] was the textbook.

Students' Backgrounds

Twenty students enrolled in the class, mostly from computer science, although a few were human-factors majors. While a few had either taken the undergraduate scenario-based design introduction, most had taken at least one graduate "usability engineering" class.

The Project

Like class 1 and class 2, there was a team-based semester-long project. Each project was site-specific. Sites selected were study lounges, parking lots, and the entire computer science building.

Coursework and Team Projects

Again, a methodology similar to the one used in class 1 was used to create project teams: students were surveyed and placed in one of four categories based upon primary skill-set, interests, available resources, and "presentation". Teams were self-organized, but had to have participants from each of the three categories.

Class Flow

The class met two times weekly for 90 minutes. The team projects were developed outside of class hours; however, four official reviews were placed on the course calendar.

SOME METHODS USED TO TEACH DESIGNING

Since it is important for pedagogical reasons to communicate the idea that there are process trade-offs, the methods selected for discussion in the classes were not arbitrary or random, but of potential utility, of some currency contributing to the language of interactive design, or intentionally provocative. Methods fell into many categories: ideation, evaluation, representation, reflection, experience design, acquiring resources, and teamwork, to name a few.

There are many methods that are considered by some of their promulgators and practitioners as fundamental to designing, in fact, *as defining what it means to design*. As part of this "just a method" stance, the underlying theory, aesthetic, and value system was addressed, however briefly – but never so much that it implied strong commitments to it. We also looked at order of presentation in the course, speculating that learning some methods might influence subsequent understanding of others. The methods listed here are the examples of ones used in the three classes.

Brainstorming

Brainstorming did not need to be taught since most of the students had been exposed to it in one way or another since elementary school. This ubiquity and familiarity also made it iconic of methods in general and how we were using them in the courses.

Design-as-Theater and Bodystorming

Ideation was encouraged through informal play-acting of situations and improvisational use of objects.

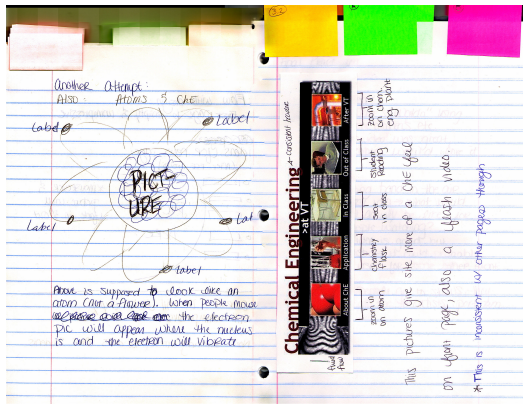


Fig 1. Sample page from a student design journal. Colored Post-Its index pages for “final exam” reflections.

Design Journals

Students were encouraged to keep journals. These notebooks are not just class notes, but repositories of ideas and observations. See Figure 1. Students were shown examples of systematic annotation to keep separate reflections that exposed their current perspective. Format was not specified, so some used paper, others laptops. This is the only method that did not include externalization for purposes of sharing with their project teammates. (This also evolved into an assessment tool that is discussed later.)

Morphological Box

Some methods were introduced with only very brief discussion. Only a few minutes was spent first introducing the morphological (or Zwicky) box [25, 33], but it was subsequently referred to repeatedly throughout the classes. As shown in the simple example in fig 2, the method is to identify independent aspects of a design, enumerate all possible variations of each aspect, and array them so as to provide every possible combination of variants. When created using rigorous criteria, it is said to define the design space. The method generates ideas since customarily rejected combinations (e.g. “buttered, beef on pumpernickel”) are created. The method also can create a combinatoric explosion of alternatives requiring time and/or methodical evaluation. It also is deceptive in that it suggests the combinations are the true extent of all possibilities. A few students had been introduced to the morphological box method in other engineering classes and often saw it as representing one of the steps in other methods. As an accessible representation as well as a method, it came to stand for the idea of design space. The box became iconic of generative design methods and the idea abstracted to any “combinatoric” design as the semester progressed.

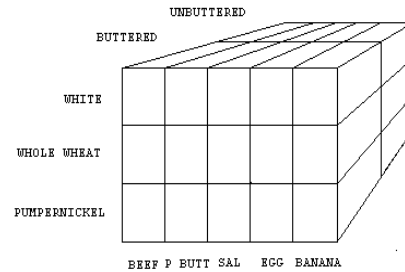


Fig 2. A morphological box illustrating the design space of sandwiches.

“Periscope” Method of Brainstorming

Because brainstorming is ubiquitous, it is possible to extend it: merging the free-wheeling, democratic, and uncritical imagining of brainstorming with the morphological box led to what we nicknamed the “periscope method”. Loosely, after selecting the most promising, novel, or clever idea from the brainstorming ideation, the participants would move up a level of abstraction and enumerate other solutions in the same space. Sample ideas would be brought down to the same level for evaluation and selection. This would test both the problem/solution space and the particular ideas generated in them. Working with this method often occurred in informal discussions and one-on-one crits. As a result, combinatoric methods – not just the morphological box -- eventually became associated with brainstorming.

Sketching and Other Representational Methods

A number of class meetings focused on methods of representation that are necessary for exploring ideas: sketching, cartooning, role-playing, foam core models, Flash presentations, and the like. As Fallman [12] and others argue, sketching is the prototypical design method. The concrete nature of these methods makes them more accessible and more obviously part of designing -- and less an intellectual word-based exercise. Students threw themselves into using hot glue guns and knives to “sketch” in foam core board during one exercise; the results littered the cramped lab space for the remainder of the semester, but were often referred to in later discussions on form and process.

Delphi Method of Project Planning

One of the first meetings of each class was devoted to teaching the Delphi method of project planning. Well known and widely practiced in many engineering and design realms, the method uses the aggregation of individual experience and expertise to develop a comprehensive project plan. In its most basic form, each member of the planning team is asked to factor a project into tasks and estimate the time each task will take. The team then negotiates differences of opinion about estimates, and sequences the tasks to form a critical path, showing dependencies and time estimates. The results of the teams were compared and this process cited as a form of reflection. This method inserted the language of management into the class discourse.

User-Centered Design

User-centered interface design [23] is the method that the majority of students had the most prior familiarity and therefore potential facility with. User-centered (and scenario-based) design is usually taught cookbook fashion, with student teams being aligned with real user communities and applications, such as accessibility-challenged users or uses of wireless technology for retail spaces. The courses immersed the students in the process by structuring homework and course presentations to take them sequentially through needs-finding, problem-finding, brainstorming, concept selection, simulation or prototyping. It was a revelation for some students to hear user-centered and scenario-based design referred to as “a” method – and not the only method relevant to interactive design.

Scenario-Based Design

One individual homework exercise had the students report a scenario of existing use [26] and rework it with an imagined new technology or new design. They were asked to reflect on how their choice of presentation media affected what they reported and more importantly, how it shaped their imagined revision. Most were familiar with the scenario method and it was spontaneously used in a number of later presentations.

Engineering (or Reductionist) Design

Most students were very familiar with the basic method of engineering: find (or be assigned) a problem, describe the problem space, describe the constraints, describe the solution space, and then optimize the solution. The ubiquity of this method often echoed in the language of the students in which “optimal solutions” were considered as “better.”

Genre Analysis

Even though the challenge was to move the students out of an engineering frame of reference, very few of the methods were from non-engineering design disciplines. The most notable was genre enquiry. The method is one that looks at systems of communication as socio-technical complexes of producers (authors, editors, designers, and publishers), media, content, consumers (readers), and setting. In the case of interactive design, the method is to look at similar situations and substitute. The assumption is that the form of the device, the appearance of the interface and kinds of interaction are communicative acts. From a traditional HCI standpoint, “user” is replaced by “reader,” placing less focus on how interaction is conveyed than how meaning is created. [5, 14]

While presented as an analytic tool, genre too became associated with combinatorics as generative method. E.g. “What would a PDA-based mystery novel be?” It had the property of being superficially analogous to some of the more phenomenal methods in HCI (such as those that use ecological or Gibsonian psychology aka “affordance analysis”) and therefore accessible.

Chance Methods

While not central to the project, methods for structuring chance (such as tossing the I Ching) were also described and demonstrated. Like genre analysis, these methods were at variance with engineering methods, values and discourse, but more so since they were not usually employed even by other disciplines found on interactive design teams. Such methods were presented as having a legitimate place in design to give permission to comfortably discuss things truly “outside the box.”

Methods Mentioned but Not Taught

A variety of other methods came up during class discussion, some were familiar to some students: pattern language [4, 11, 34], ethnographic observation, mind maps, reflection, participatory design, cognitive walkthroughs and even extreme programming [6]. Other similar introductory design classes have used mind maps and semi-structured design journals explicitly, but were only introduced as methods that might be adopted. Again, *all were presented as just methods*.

REFLECTION, REINFORCEMENT, AND ASSESSMENT

Introducing a smorgasbord of design methods is not an end in itself; it must be balanced by reflection and thoughtful critique about process and products. To create dialogue, to legitimize the value of anticipating and planning design process, and to develop some shared understanding and common ground, critique must be seen as the primary driver of design iteration³.

In the courses, reflective thinking was often used rhetorically as a test of legitimacy of decisions. People would say, “Thinking about what we did...” or “I’m not sure that this is really worth the time to talk about...” (This latter prefaced a reflection on the value of reflection in a particular situation.)

While some guidelines such as “suggest alternatives instead of just criticisms” were provided and at times adopted, criticism was a difficult experience. The most difficult part of critique is accurate hearing and understanding between designer and reviewer. This is best illustrated by the interim design reviews.

Official Reviews

Interim Review Sessions

At each of the in-class review sessions, the teams presented PowerPoint presentations with various mockups and prototypes of ideas and a scenario of use. The rest of the class was invited to participate in the critique of each presentation. Other students and outsiders (such as other CS faculty) who happened to attend provided a great deal of technical review and would suggest fixes and alternative implementations.

The feedback took the form of value-driven comments that were intended to keep students dissatisfied with their de-

³ Rather than refinement or additional depth of a particular design.

sign solutions. For example, we might say, “You haven’t given this enough thought yet.” “The ideas are not well fleshed out.” Or “This is a good start, but you need to make this real.” And the student designers would respond with “This is a great solution to that problem.” To which we might say, “But is that really the right problem?” There were many variations of this dialogue in informal settings, as well.

Interim Review Written Critiques and Responses

Interim reviews also had structured written critique and response. Each project would be assigned peer reviewers who would write three or four pages of critique; a copy would go to the project team and another to the faculty for grading. The project team would then write a response that would include possible revisions to the project design and possible methods that could be used to address the critique. Subsequent project reviews were required to refer to the written reviews and responses.

Informal Reviews and Project Discussion

The point of the interim review sessions was to encourage students to break away from their engineering-centric training that led them to seize upon and solve surface-level specific problems rather than the deeper ideas and larger issues behind the problems. Part of this also involved the adaptation of the students to the need to perform well in a project class: define a problem quickly and solve it quickly, then move on. We have already noted that the methods that students already knew – reductionist engineering design, user-centered design, for example – were used as rhetorical foils. Discussions of immediate problems and alternatives to unsatisfactory results were often put in terms of reflecting on the de facto method. In other words, where the student designer argued that they had created an optimal solution, we would respond with a discussion of their choice of method: the point being to break free of the one problem/one solution mindset.

Formal Reflection on Designing

To further reinforce the pattern of critique and reflection and look beyond the topics of the team projects, the last few class sessions of the graduate classes (1 and 3) were given over to a collective review of articles in *The Idea of Design* [18]. The book is a compilation of wide-ranging articles from *Design Issues* and other sources on the nature of design, the meaning of products, and the relationship of design and culture. The review took the form of a familiar graduate teaching method (assigning chapter to selected students for group discussion). Unlike a standard seminar, the students were asked to interpret their project experience in terms of the article.

This morphological approach was itself again labeled as “just a method”. Therefore, one student discussed Csikszentmihalyi’s notion of flow, where it occurred in his explorations of the theme and how he anticipated his users experiencing it. Another, selected to read and report on an article titled “Product Symbolism of Ghandi and Its Con-

nection with Indian Mythology” rethought the cognitive aspects of the interaction as a product of the engineering culture he was educated in. The use of the methods rhetoric in this more intellectual setting not only *applied practice to theory*, but provided a larger reflective frame for the graduate pedagogy.

From Methods To Meanings

The projects required that the final products “speak with one voice”. This is generally challenging for students. Reviewing which methods had been used was a means to move discussions from whether an optimal implementation had been created to wondering how others perceived the system, to assessing the system’s meaning more generally. This review was accomplished in part by asking whether a genre methodology would have produced the same result, for example, or if a comparison of foam core mockups revealed any differences. Getting from the wide discourse about designing enabled by the “just a method” rhetoric to student internalize a wider view of the role of designers in making meaning was in this way straightforward.

Measuring Success

Following the remarkable set of projects that emerged in Class 1 (see Results below), design thinking, reflection, and assessment were merged in Classes 2 and 3. Keeping design journals was required. A final “exam” asked students specific questions that could only be answered by using excerpts from their journals. (In Class 2, students were instructed to create an information architecture – a major topic in the course -- in the week prior to the exam as an aid to completing it.)

Using Post-It notes to index answers, they wrote responses to questions like: *Describe an idea that was rejected by your team for the semester project. Explain why it was rejected and – given how your project turned out – whether that was a good decision.* All questions were designed to both elicit demonstrations of design thinking and to highlight for the student what design thinking occurred.

For this paper and to suggest refinements to the method, the journals and exams were again reviewed looking for references to specific methods and for reflections on methods in general. While the great majority of journals had one or two references to particular methods, few had reflective comments tying methods together. In contrast, about half of the exams made mention of the value of structuring processes. These reflections occurred both in response to questions about project process and in ones about project result. As optimistic teachers, we see this as initiating a pattern of design thinking; as realistic teachers, we also see this thinking at risk in other courses and entry-level employment.

THE IDEA OF METHODS

Since the objective of the class was to introduce students to comprehensive design, it was clearly and repeatedly stated that students were not being evaluated on the use of methods, per se and that the classes were not survey classes in

HCI methods. Methods were presented as a means to understand a design topic and a potential tool to employ as their projects unfolded. Students were told, however, that they should be prepared to defend their decision to use or not use a method.

Design Process And Use Of Methods

"Resistance is futile!"

There is a natural resistance to the use of methods. There are a number of very good reasons for this: lack of trust in the method, a resistance to taking orders, a belief that the answer is already known, ego, power relationships, or that there is a better way. The student-designers used all of them to resist using some of the process methods.

For example, one student had been creating very dense mind-maps of problem spaces and solutions using an idiosyncratic coloring scheme for a number of years before he came into the class. Beautiful to look at and difficult for others to decode, it was a method he was very attached to. While this blocked him from adopting other methods, it did not block discussion of what the method provided him, its underlying principles, and how well it worked with others (particularly as a tool for communicating with others).

Friction, Resistance, Synthesis (Sometimes)

At times the process was dialectical: students would push back on the relative value of one method or the other, and a new common understanding would emerge. Other times there would be no obvious synthesis that would result from the process. Framing the process in these terms is useful to convey some sense of the experience, but distorts the experience of designing the projects in a couple of ways.

Friction and resistance were often not central to the evolution of the artifact or the quality of its realization. In all classes, there were debates about whether the Delphi or any method for that matter could actually predict product development. Even though these debates did not reach consensus, they would continue throughout the semester as a shared experience and point of departure for other issues surrounding individual contribution to team efforts.

RESULTS

The careful reader will study the pictures and note these are not the expected work of students given the same problem statement and asked to create the optimal solution.

Class 1

The four projects that emerged were SeismoSpin (fig 3), Shazam or [Sh@z4M](#) (fig 4), Eeeww! (fig 5), and LOUD (fig 4).

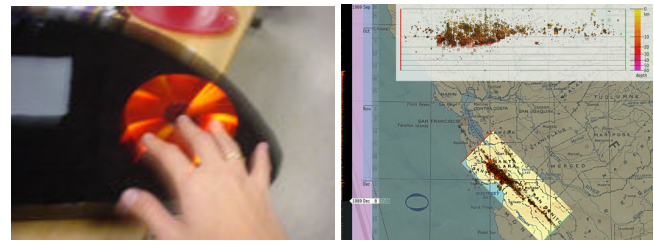


Fig 3. Left: SeismoSpin uses a DJ's "scratching" interface to control display patterns of seismicity. Right: earthquakes are shown on a map and a sectional image. Spinning the disk moves backwards and forwards in time; a lever shifts the time scale from seconds to centuries; and a stylus pad selects horizontal sections to show depth.



Fig 4. Left: Shazam. A digital light show system. The controller is a chalkboard so that buttons can be annotated with notes or drawings of effects associated with them. The TacTex pad works and a rhythm wand (not shown) are used to set beat and modulate visual effects. Right: LOUD is a selection interface to a library of music. It is intended to aid in the identification of genre, performer, beat, instrument, and other aspects by rotating in the chair to face one of four different channels.

This class was a lot of hard work for the students. Besides the typical late-night sessions towards the end of the semester, coordinating grad-student schedules made design meetings hard to arrange during the bulk of the semester. Everyone had some sort of gripe: teams couldn't make decisions, some people felt that other members of their team were not pulling their load, resources were scarce, the work space was bad – the list was as one would expect. Still the course projects were well received by the design jury and received subsequent publication in several venues (two short papers at CHI, AIGA's LOOP journal, and UC Berkeley's Lab Notes column) [9, 21].

Class 2

The ten teams produced a range of website designs. The most comprehensive and creative is a massive multiplayer on-line game. Freshmen wanting to find the inside "skinny" on a BS in Mechanical Engineering take on a role and explore a virtual ME Department. Bots can answer questions about the department by mining official departmental and course websites. More traditional approaches for other departments included narrative vignettes describing hurdles in "weed-out" classes, social functions, and aspects of lab work. All of this was conceived, produced and evaluated by student teams whose prior experience was standard undergraduate computer science curriculum.

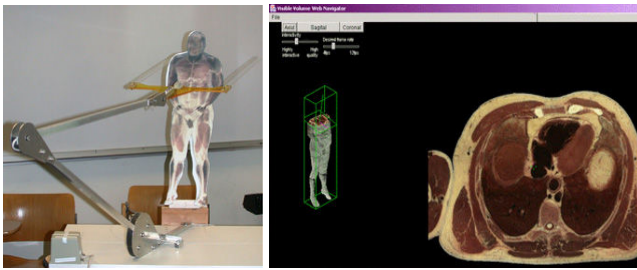


Fig 5. Left: Eeeww! cut sections through a digital on-line cadaver. Right: Eeeww! used a repositionable wand to show where the virtual slice is made. It is intended for middle school human biology classes. The project name was suggested by users brought in to evaluate the project.

Class 3

Most recently, the teams in the Design of Interactive Systems course developed four site-specific interventions based upon problems the teams uncovered. Two examples (shown in figures 6 and 7) show the range of design exploration.

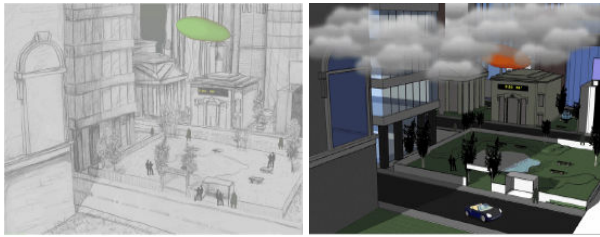


Fig 6. Study lounge public display. Objects would come and go, day would change to night, even the style of rendering would change – all linked to events in and about the space. Awareness of others with common interests would be signalled with particular characters, bus arrivals and departures just outside the lounge would be noted by a bus on the street, and changes in weather would also change the appearance of the scene. Quite intentionally, the student designers provided no key to the semantics, thus requiring study of the scene to uncover patterns and encourage social interaction in the study lounge.

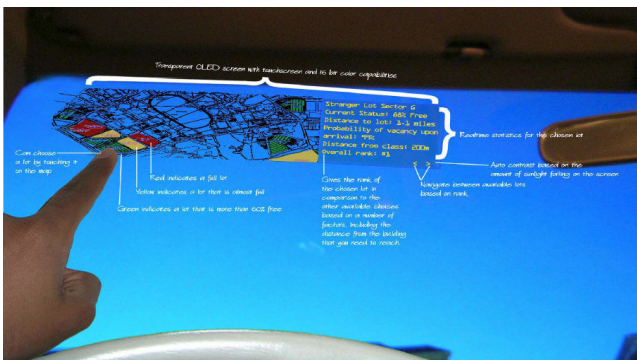


Fig 7. pPNG is a system to find a parking place for routine commuters. In-vehicle display comes on when approaching lots.

One project put a slowly changing image of a town square on a public display in a very popular study lounge. (Figure 6.) Another was a comprehensive parking information system that provided lot availability information before depart-

ture and would direct subscribers to the nearest available slot.

DISCUSSION

Using a combination of design methods and doing more or less continuous critique, we tried to get students to understand that designing is thinking widely, critically, and cleverly. While all the methods taught had some relevance to current purposes, it did not matter particularly what methods we taught; it did matter that we called out many of the things they already knew and mistook for canonical design as "just another method". We engaged in a rhetorical situation that legitimized questioning design direction. What does this suggest about further design research and design pedagogy research?

From the very first meeting, students were asked to think and re-think: "What is design? How do design representations affect design thinking? How is meaning created? What is the difference between user and consumer or reader and user? What is a reflective practice? How do we work together? How do we decide what is the right thing? What is under my control? How do I design the process I am engaged in?" And most importantly, "What have I done?" and "What should I do next?" All of these questions were made the students' own by the use of the *idea* of methods.

Methods in Other Forms of Design Education

The basic program in architectural design education is a four year B.A. followed by a two year Masters program. In almost every semester of those six years, a student will take at least one project-based design class – that is, a total of about 10 to 12 design classes. While it is rare today for design methods to be taught as a separate subject, the accumulation of experience with different instructors with different approaches to design and different project types exposes the architecture student to a variety of methods. It is this accumulation and the informal link from one class to the next that provides the framework for reflection. In contrast, most HCI design is taught in just one or two classes, so there is a need to compress this and make it as visible as possible.

Beyond Design Education

Practicing designers work with tried and true methods. In fact, with habituation and expertise may come the loss of recognition that there are separate activities or even that the act of design consists of methods. Yet even experts may benefit from decomposing their root thoughts. We agree with Schon in advocating reflection, but where Schon advocated reflection upon a unified notion of activities, process and outcomes, the current work suggests that reflection upon the *labeling* of activities may also lead to better, more appropriate design activities and more self-aware designers.

Methods in Science, Methods in Design

We have been implicitly arguing that researchers and designers see and use "methods" in very different ways from each other. Scientific investigation does not and would not

employ methods that are at variance with underlying principles. Designers have no problem doing just that if it solves the problem at hand. Each view of “methods” is correct in its own realm; however, this fast and loose treatment of theory can raise ethical concerns: for example, when designers use ethnographic methods for requirements gathering, anthropologists correctly cringe since designers do not usually possess the rigorous training or take the extensive time to reflect upon the patterns of meaning that would be expected of serious scientific enquiry. The instrumental drive of design should not be a license to use any process for any purpose. Therefore, it is essential that designers understand *reflection is not just a method, but an underlying principle*. At one level, reflecting on process is the ethical cost of pragmatic use of methods from very different paradigms. At another level, reflection is the essential integrator of knowledge.

Beyond Design

In the introduction, we recognized the necessity for designers to gain an awareness that designing is composed of activities, that members of a design team must build shared understanding of the constituents of each activity, and that designing – like artifacts – has the potential for being designed. In other work situations, we see the necessity for reformulation of activity – particularly as collaborators become distributed in space, time and organization and their activities mediated. The work reported here is specific to design pedagogy but suggests that the ways in which activities are “parsed” from everyday experience and their labels negotiated in collaborative work may influence how well work is established and reformulated in improvisational settings.

CONCLUSION

The relationship between acquiring and developing deep domain knowledge and acquiring and developing broad general knowledge (particularly in and about social settings) should be better understood. This paper is just one take on it. Pedagogical experimentation should focus on not just broadening future interactive designers’ current knowledge, but on design methods and practices that create ongoing breadth and perspective. Further studies should also address the role of methodological rhetoric as a meta-method investigating what promise *the idea of methods* holds for design.

ACKNOWLEDGMENTS

The authors would like to thank John Canny, Carlo Sequin, and the students who took CS 294-12, CS 4634, and CS 5984. We would also like to thank *alt.chi* (2005) who selected an earlier version of the paper and allowed us to refine the message of this version.

REFERENCES

1. ... Berkeley Institute of Design. (2003) <http://bid.berkeley.edu>
2. ...Design Realization/Berkeley Institute of Design. (2003) <http://bid.berkeley.edu/bidclass>
3. ... user interface design/Computer Science, UC Berkeley. (2002) http://guir.berkeley.edu/courses/cs160/2002_spring/
4. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., and Angel, S.A. (1977) *A Pattern Language*. New York: Oxford University Press.
5. Back, M. and Harrison, S. (2002) The Roads Not Taken: detours and dead ends on the design path of Speeder Reader. *Proceedings of DIS '02*, ACM Press, London.
6. Beck, K. (2000) *Extreme Programming Explained*. Reading MA: Addison Wesley.
7. Bjerknes, G., Ehn, P. and Kyng, M., eds. (1987), *Computers and Democracy - A Scandinavian Challenge*, Avebury.
8. Crabtree, A. (2004) “Design in the absence of practice” *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques* Cambridge, MA, USA pp 59 - 68
9. De Guzman, E., Back, M., Harrison, S., Ho-Ching, W., Matthews, T., Rattenbury, T. (2003) “EEWWWW! Tangible Interfaces for Navigating Inside the Human Body.” *Proceedings of CHI 2003, Short Papers*, Ft. Lauderdale, FL, ACM Press.
10. Dourish, P. (2001) *Where the Action Is: the Foundations of Embodied Interaction*. MIT Press, Cambridge.
11. Erickson, T. (2000) Lingua Francas for Design: Sacred Places and Pattern Languages. *Proceedings of DIS '00*, ACM Press, New York, pp 357-368.
12. Fallman, D. (2003) Design-oriented Human-Computer Interaction, *Proceedings of CHI2003, Conference on Human Factors in Computing Systems*, (Fort Lauderdale, Florida, April 5-10), New York, NY: ACM Press, pp. 225-232.
13. Harrison, S. (1993) Computing and The Social Nature of Design. *ACADIA Quarterly*. University of Southern California, Los Angeles; v.12, no. 1, Winter, pp 10-18.
14. Harrison, S., and Minneman, S. (1995) Studying Collaborative Design to Build Design Tools. *The Global Design Studio. Proceedings of the Sixth International Conference on Computer-Aided Architectural Design*, September 24-26, Center for Advanced Studies in Architecture, National University of Singapore. pp 687-698
15. Harrison, S., and Minneman, S. (1996) A Bike In Hand *Analysing Design Activity*, (Cross, Christians, & Dorst, eds.) Chichester, UK, John Wiley & Sons pp 417-436.
16. Jones, J. C., (1992) *Design Methods*. Second edition, New York: Van Nostrand Reinhold.

17. Lowgren, J., Stolterman, E. (1999) Methods & tools: design methodology and design practice, *Interactions*, v.6 n.1, p.13-20, Jan./Feb.
18. Margolin, V., Buchanan, R. eds. (2000) *The Idea of Design*. Cambridge: MIT Press.
19. McCullough, M. (2004) *Digital Ground. Architecture, Pervasive Computing, and Environmental Knowing*. MIT Press, Cambridge.
20. MacKay, W. (2004) "The interactive thread: exploring methods for multi-disciplinary design" *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques* Cambridge, MA, USA pp 103-112
21. McKelvin, M., Back, M., Harrison, S., Nestande, R., Valdez, L., and Yee, K. (2003) "SeismoSpin: An Interactive Instrument for Seismic Data." *Proceedings of CHI 2003, Short Papers*, Ft. Lauderdale, FL, ACM Press.
22. Moran, T., Carroll, J. eds. (1996). *Design Rationale: Concepts, Techniques, and Use*. Lawrence Erlbaum Associates, Hillsdale, NJ.
23. Norman, D., Draper, S. (1986) *User Centered System Design; New Perspectives on Human-Computer Interaction*, Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
24. Pahl, G., Beitz, W. (1998) *Engineering Design: A systematic approach*, Springer.
25. Ritchey, T. (2002) "General Morphological Analysis. A General Method for Non-Quantified Modeling" <http://www.swemorph.com/ma.html>
26. Rosson, M., and Carroll, J. (2002) *Usability Engineering. Scenario-Based Development of Human-Computer Interaction*. San Francisco: MorganKaufmann Publishers.
27. Schon, D. (1983) *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.
28. Schraefel, M., Hughes, G., Mills, H., Smith, G., Frey, J. (2004) "Making tea: iterative design through analogy" *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques* Cambridge, MA, USA pp 49-58
29. Simon, H., (1996) *Sciences of the Artificial – 3rd Edition*. Cambridge, MA: MIT Press.
30. Smith, H., Fitzpatrick, G., Rogers, Y. (2004) "Eliciting reactive and reflective feedback for a social communication tool: a multi-session approach" *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques* Cambridge, MA, USA pp 39-48.
31. Stults, R., Harrison, S., and Minneman, S. (1989) "The Media Space – experiences with video support of design activity. In *Engineering Design and Manufacturing Management* (Samuel, A. ed), Amsterdam: Elsevier, pp 164-176.
32. Suchman, L. (1987) *Plans and situated actions: The problem of human/machine communication*. Cambridge University Press, Cambridge.
33. Swedish Morphological Society (2004) <http://www.swemorph.com/>
34. van Duyne, D., Landay, J., and Hong, J. (2003) *The Design of Sites. Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Addison-Wesley, Boston