"It's Just a Method!"

A Pedagogical Experiment In Interdisciplinary Design

Steven Harrison
Department 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

ABSTRACT
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 [10, 26]. Yet, as Suchman argued long ago [28], planning accounts may be applied to problems that are not at base accomplished by planning. This paper reports on a method that takes Suchman's criticism to heart and avoids dressing up design methods as more systematic and predictive than they in fact are. This method focuses on the teaching of methods as a means to engage in reflective critique. It is not about any one design method – or even a suite of methods – but the *idea* of methods as a means to understand what is and what is not under the control of designers.

The paper reports an effort at this reframing in a graduate team-based design class. While the paper reports an early investigation in the pedagogical application of the idea, the direction suggests underlying factors that may explain the effects of other, allegedly more "systematic" methods.

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 methodologies.

INTRODUCTION
Interaction design is at a major decision point. Recent books from Paul Dourish [9] and Malcolm McCullough [17] not only point to changing paradigms of human computer interaction, but to new paradigms of designing human computer interaction. The approach of the old HCI designing paradigm was to make designers into engineers. To name one example, Rosson and Carroll [24] have through their work on the case method tried to develop the capacity of their students to be better observers and analysts. This aligns with a major pull in the field of design over the past thirty years, stemming from Simon's notion of a design science [26].

Designers need to be able to reform their processes to accommodate the possibilities inherent in embedded and emergent activities. Young designers need to gain and regain a flexible ability to see activity at various granularities. In particular they need experiences that give them 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 an artifact – has the potential for being designed.

We start from two structuring premises: that design is making something new that fits with reality; and that "design activity is importantly social since it involves working with and for others to create things of value and utility to others."[1]

The first premise leads us to the nature of design knowledge. Probably the best general characterization is that it is wide and that design schemas are fluid. This seems to be true for all design disciplines, even ones such as software that also require deep domain knowledge. (That is, successful designers seem to have wide, inter-domain knowledge as well as deep intra-domain knowledge.) When looking at both process and results, the design skill that this enables is the ability to reconceptualize. When looking at leverage points in design processes, reconceptualizing is tied to the observation that problem finding is as important as prob-

CHI 2005, *APRIL 2–7, 2005, PORTLAND, OREGON, USA.*
ACM 1-59593-002-7/05/0004.

[1] Originally in [25], but echoed in part by others: Linus Torvalds, "we strongly believe that to do something really well, you have to get a lot of people involved." (NY Times Magazine, Sep 28, 2003, p. 23.)

lem solving. One response to this is to create multi-disciplinary teams. However, this often leads to conflicts around role/discipline-based responsibilities and the need to find common language among team members. Another common response is to broaden the expertise of designers, to feel comfortable thinking across disciplinary boundaries.

The second premise builds on much research on design process over the past twenty years that has focused on the interactions of designers and the development of tools to support those interactions. [12, 13, 14, 16, 27] If intellectual structure provides one inroad for training, discourse practices and habits provide another.

Most recently, we applied some of those characterizations and concerns to issues in teaching a graduate course in the design of human-computer interaction. Students came into the class with strong backgrounds in software engineering. 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. The course had a number of project reviews during which feedback was given on the designs; the reviewers (both instructors and outsiders) gave value-driven comments that on the whole were intended to keep students dissatisfied with their design solutions; the reviews were not up/down or pass/not pass, but were part of a dialog between design and critique. Also, students were immersed in the language of many design methods, not just one or two.

A byproduct of this process was to encourage the students to become part of the community of interactive systems designers, learning the language and approach of the discipline. Most importantly, since feedback outside of traditional software engineering or HCI analysis was treated as legitimate, they broadened the terms in which they understood the notion of problem definition and problem solution. In particular, they worked *with* a variety of methods to work *through* the methods into richer, multi-layered meaning in the final design.

PREVIOUS WORK

Human-computer interaction is awash in methods and the theories that underlie them. However, there is a limited amount of research on the value to the practice or learning of designing about the place of methodologies, *per se*, in this discipline. There are two works that look at design in general which suggest the utility and role of meta-methodology: John Chris Jones' *Design Methods* [15] (particularly the second edition with its reflective and extensive introductions) and Don Schon's *Reflective Practitioner* [26].

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 itself a method, of course. It also raises to the level of meta-method since it incorporates the idea of reflection upon the act of method and utility of reflection.

Many methods and their underlying theories have advocates and practitioners who see great value in adopting them as approaches to design. For purposes of this paper and the teaching approach, we abstract the method away from the underlying theory or from any specific approach.

The approach of participatory design is probably the closest to the course's in that it calls upon the designer to find the appropriate method with which to listen to the user [7]. Further, in subsequent writings, Bjerknes and Ehn have written about the social practice of design among interdisciplinary design teams where the reformulation of methods for specific partner-users in specific contexts.

While the meta-method of participatory design 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 [22]. Closer to home, design rationale [20] 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 [11] 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.

BID

The class was a prototype of an anticipated core class for a proposed program in interactive design. The Berkeley Institute of Design [1] is intended to use a multidisciplinary and designerly approach to create innovative technology. This is in contrast to most of the existing research-oriented graduate study in human-computer interaction. Quoting from the mission statement at the time:

> The Berkeley Institute of Design (BID) is a proposed research/teaching unit that fosters a new and deeply interdisciplinary approach to design for the 21rst century:
>
> The design and realization of rich, interactive environments which are shaped by the human activities they support.
>
> Here "environments" include architectural spaces, products, web sites, and other artifacts that support complex human activity. The program combines technical and social/humanist perspectives on design. It acknowledges that design in the era of

ubiquitous technologies means not only technical innovation, but deep understanding of behavior and the experience that technology should enhance. It mixes engineering design with psychology, social sciences and art practice. It combines Berkeley's rigor in engineering with its commitment to social values and critical reflection.

Objectives of BID:

• Educate students on the breadth of topics that are important for 21st century design
• Develop students' skills in team-work, communication, and creativity
• Promote excellence in the practice of design within and across professions
• Expose students to real-world design problems and bringing concepts to reality
• Cultivate students' ability to express, evoke and shape experience through design
• Foster critical reflection on technology and the contexts that shape its use
• Create a generation of designers who lead product development in large companies

Broadening The Base Versus Multidisciplinary Teams

There are a number of reasons to consider broadening the knowledge and confidence of discipline experts, instead of assembling a multidisciplinary team:

- Sometimes experts in other disciplines are not available.
- Many projects cannot support the variety of disciplines required.
- The start up time required to get a team into trusting working relations with some common language may not be available.

All of these practical training reasons were complementary to the overall goals of BID and to the particular situation of this class. And, they are complementary to multidisciplinary design teams since broadening the experience of domain experts can ease working relations when projects are multidisciplinary.

Class

The course called "Design Realization" is designed to be the second core class that masters and Ph.D. students would take upon entering the BID program. It is intended to follow a class that is a general introduction to design and representation techniques. Since this was a prototype for a program not yet running, neither that preceding class nor the students for the BID program were in place.

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 cramming in 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 the SIMS program (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 SIMS 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

[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.

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.

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. 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.

METHODS

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*. It was important for pedagogical reasons (that is, to communicate the idea that there are process trade-offs) to present the idea that each is "just a method". (See the discussion at the end for the practice ethos and methodology that extends from this.)

So the methods selected for discussion in class were not arbitrary or random, but of potential utility, of some currency contributing to the language of interactive design, or intentionally provocative. As part of this "just a method" stance, the underlying theory, aesthetic, and value system was addressed, however briefly. 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 primary ones we covered, in discussion if not necessarily in practice.

Delphi Method of Project Planning

One of the first class meetings 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 class was arbitrarily divided into three teams and asked to plan the process of designing a simple interactive device. Each team had to complete this cartoon problem and the results were compared. Since this was before term project teams had been formed it also gave the students some experience working together.

The results of the three 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

One of the next methods discussed was actually a pre-requisite – user-centered interface design [3]. A one semester class, it is the method that the majority of students[3] had the most familiarity and potential facility with. The class is usually arranged cook-book 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 course immerses 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. Therefore, in the design realization class, these process steps and the overall goal of user-centered design could be incorporated as part of the designing vocabulary from the very beginning.

It was a revelation for some students to hear user-centered design referred to as "a" method – and not the only method relevant to interactive design. In part, the research theme of "instruments" had been chosen as an essential element of the term project in order to break the idea that users' pre-existing skill could be the measure of success.

Engineering (or Reductionist) Design

Another method that was assumed most all knew well is the basic method of engineering: find problem, describe the problem space, describe the constraints, describe the solution space, and then optimize the solution. Most often this is structured with a problem being given which the student-engineer needs to solve optimally. The ubiquity of this method often echoed in the language of the students where "optimal solutions" were considered as "better".

Morphological Box

Some methods were introduced with only very brief discussion. Only a few minutes was spent first introducing the morphological (or Zwicky) box [23, 29], but it was subsequently referred to repeatedly throughout the class. As shown in the simple example in fig 1, 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.

---

[3] As with many classes, a few enthusiastic students were admitted without the pre-requisite.

It also is deceptive in that it suggests the combinations are the true extent of all possibilities.

As an accessible representation as well as a method, it came to stand for the idea of design space. 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. The box became iconic of generative design methods and the idea abstracted to any "combinatoric" design as the semester progressed.

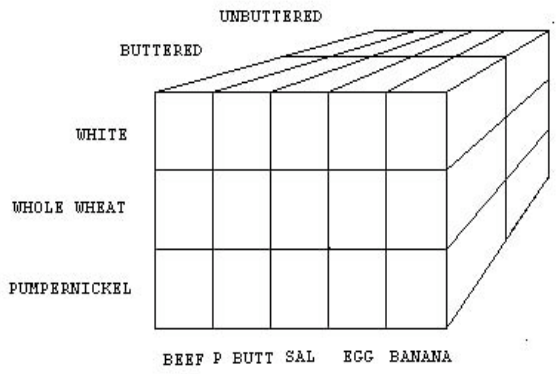

Fig 1. A morphological box illustrating the design space of sandwiches.

"Periscope" Method of Brainstorming

The combinatoric method eventually became associated with brainstorming. Brainstorming as method, like the engineering method, did not need to be taught since most of the students had been exposed to it in one way or another since elementary school.

Because the method 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.

Scenario-Based Design

One individual homework exercise had the students report a scenario of existing use [24] and re-work 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. Many were familiar with the scenario method and it was spontaneously used in a number of later presentations.

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" which means less a focus on how interaction is conveyed than how meaning is created. [5, 13]

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."

Representational Methods

A number of class meetings focused on the "little"[4] methods of representation that are necessary for exploring ideas: sketching, cartooning, role-playing, foam core models, Flash presentations, and the like. 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 class; 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.

Methods Mentioned but Not Taught

A variety of other methods came up during class discussion, some were familiar to some students: pattern language [4, 10, 30], ethnographic observation, mind maps, reflection, participatory design, cognitive walk-throughs 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

[4] By "little" we do not refer their centrality or periphery to designing. As Fallman [11] and others argue, sketching is the prototypical design method.

methods that might be adopted. Again, *all were presented as just methods*.

CRITIQUE AND REFLECTION

Introducing a smorgasbord of design methods is not an end in itself; it must be balanced by reflection and thoughtful critique. To create dialogue, to legitimatize 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.

In the course, 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 preface being 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

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.

As instructors, we critiqued the state of their project. 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.

Informal Reviews and Project Discussion

The point in 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 were given over to a collective review of articles in *The Idea of Design* [17]. 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 Connection 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.

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 many of the process methods.

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. There were debates about whether the Delphi or any method for that matter could actually predict product development. Even though the class never came to a shared resolution of the debate, it did continue throughout the semester as a shared experience and point of departure for other issues surrounding individual contribution to team efforts.

From Methods To Meanings

The project required that the final product "speak with one voice". The design of the enclosure, the means of interaction, the database that held the content the instrument would play, the message that the overall object conveyed and the imagined setting of use had to be designed together. Getting from the wide discourse about designing enabled by the "just a method" rhetoric to having the students internalize a wider view of the role of designers in making meaning was straight-forward.

Reviewing which, if any, 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. This was accomplished by asking if a genre methodology would have produced the same result, for example, or if a comparison of foam core mockups revealed any differences.

Following the results section is a discussion of both the details of the consequences of these particular methods in this particular class situation and a discussion of how reflecting on methods leads one to anticipate process and think widely about artifacts.

RESULTS

The four projects that emerged were SeismoSpin (fig 2 and 3), Shazam or Sh@z4M (fig 4 and 5), Eeeww! (fig 6 and 7), and LOUD (fig 8).

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.

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 (CHI, AIGA's LOOP journal, and UC Berkeley's Lab Notes column).

DISCUSSION

It is possible that none of this meta-methodology activity had any impact at all on the projects or the learning of design. These were talented students who were given a project that challenged them. The interim reviews and design crits pushed them quite hard. Arguably, this class might have turned out the same set of projects if we had never said, "It's just a method". However, the students would never have developed such a rich understanding of their own and others' design processes -- nor such deep rationales for the final forms of their work.

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

Fig 2. SeismoSpin uses a DJ's "scratching" interface to display patterns of seismicity. Selected for a CHI short paper [19] and the AIGA LOOP Award.

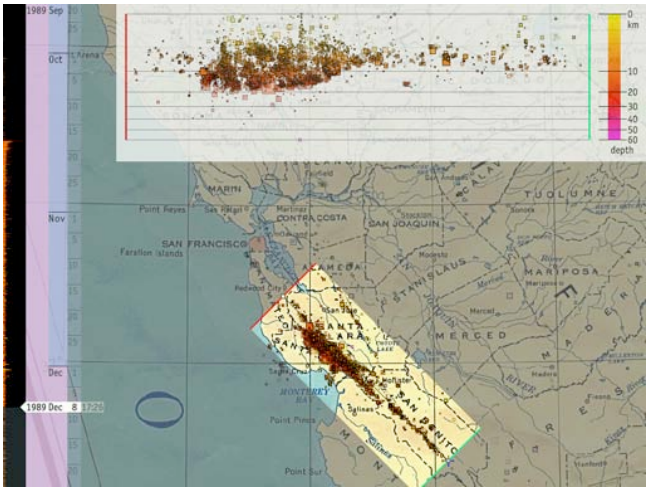

Fig 3. SeismoSpin's display: earthquakes are shown on a map and a sectional image. Spinning the disk moves backwards and forwards in time; a lever shifts the time scale from seconds to centuries; and a stylus pad selects horizontal sections to show depth.

Fig 4. 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. Selected for the AIGA LOOP Award.

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.

Readers of this report are probably steeped in research methods. It may come as some surprise, then, to realize that the world does not, *a priori*, think of work or its organizing activities as malleable units called methods. Routine, everyday action does not usually need a name and need not be thought of as an independent segmented activity. Giving analogous activities names and methods creates a reflective framework for design work.

Using a combination of teaching design methods and doing more or less continuous critique, we tried to get students to understand that designing is thinking widely, critically, and cleverly. 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?

Fig 5. Shazam played repeating visual effects that would be "performed" by the "visual artist" using the controller to modulate the algorithm. The handheld shaker could move the image around and in and out from the viewer. Its position was sensed by a Flock Of Birds sensor.

Beyond Design Education

Practicing designers work with tried and true methods. In fact, with habituation may come the loss of recognition that there are separate activities or even that the act of design consists of methods. Where Schon advocated reflection upon the activities and their outcomes, this work suggests reflection upon the labeling of activities, regardless of outcome, and may also lead to better, more appropriate design activities and more self-aware designers.

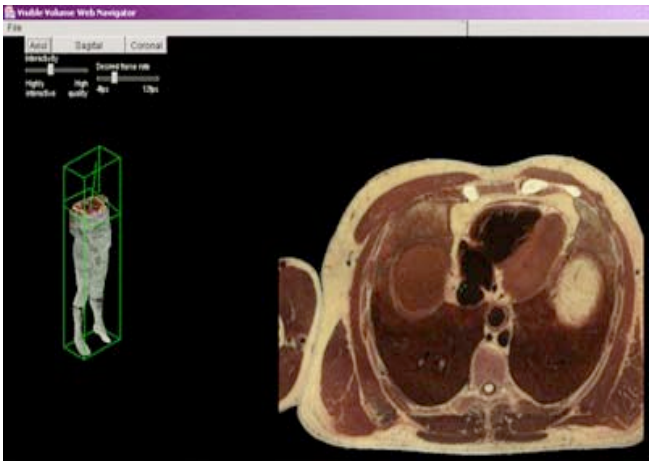

Fig 6. Eeeww! cut sections through a digital on-line cadaver. Selected for a CHI short paper [8] and the AIGA LOOP Award.

Fig 7. 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. It also used the Flock Of Birds sensor.

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

Fig 7. 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. Students hand-built a coarse rotary shaft encoder to sense the direction of the chair. The most complex system architecture among the four projects, four stereo playback channels selected from an aspect database of musical selections.

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 on-going 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 in Fall 2002.

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. De Guzman, E., Back, M., Harrison, S., Ho-Ching, W., Matthews, T., Rattenbury, T. (2003) "EEWWW! Tangible Interfaces for Navigating Inside the Human Body." *Proceedings of CHI 2003, Short Papers*, Ft. Lauderdale, FL, ACM Press.

9. Dourish, P. (2001) *Where the Action Is: the Foundations of Embodied Interaction*. MIT Press, Cambridge.

10. Erickson, T. (2000) Lingua Francas for Design: Sacred Places and Pattern Languages. *Proceedings of DIS '00*, ACM Press, New York, pp 357-368.

11. 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.

12. 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.

13. 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

14. 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.

15. Jones, J. C., (1992) *Design Methods*. Second edition, New York: Van Norstrand Reinhold.

16. Lowgren, J, Stolterman, E. (1999) Methods & tools: design methodology and design practice, *Interactions*, v.6 n.1, p.13-20, Jan./Feb.

17. Margolin, V., Buchanan, R. eds. (2000) *The Idea of Design*. Cambridge: MIT Press.

18. McCullough, M. (2004) *Digital Ground. Architecture, Pervasive Computing, and Environmental Knowing*. MIT Press, Cambridge.

19. 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.

20. Moran, T., Carroll, J. eds. (1996). *Design Rationale: Concepts*, Techniques, and Use. Lawrence Erlbaum Associates, Hillsdale, NJ.

21. Norman, D., Draper, S. (1986) *User Centered System Design; New Perspectives on Human-Computer Interaction*, Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

22. Pahl, G., Beitz, W. (1998) *Engineering Design: A systematic approach*, Springer.

23. Ritchey, T. (2002) "General Morphological Analysis. A General Method for Non-Quantified Modeling" http://www.swemorph.com/ma.html

24. Rosson, M., and Carroll, J.(2002) *Usability Engineering. Scenario-Based Development of Human-Computer Interaction*. San Francisco: MorganKaufmann Publishers.

25. Schon, D. (1983) *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.

26. Simon, H., (1996) *Sciences of the Artificial – 3rd Edition*. Cambridge, MA: MIT Press.

27. 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.

28. Suchman, L. (1987) *Plans and situated actions: The problem of human/machine communication.* Cambridge University Press, Cambridge.

29. Swedish Morphological Society (2004) http://www.swemorph.com/

30. 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