

External Representations in Ubiquitous Computing Design and the Implications for Design Tools

Steven Dow¹

College of Computing¹
GVU Center
Georgia Institute of Technology
Atlanta, GA 30332, USA
steven@cc.gatech.edu

T. Scott Saponas²

DUB Group²
Computer Science and Engineering
University of Washington
Seattle, WA 98105-4615 USA
{ssaponas, yangli, landay}@cs.
washingt.edu

Yang Li²

James A. Landay^{2,3}

Intel Research Seattle³
1100 NE 45th Street
Suite 600
Seattle, WA 98105
james.a.landay@intel.com

ABSTRACT

One challenge for ubiquitous computing is providing appropriate tools for professional designers, thus leading to stronger user-valued applications. Unlike many previous tool-builders' attempts to support a specific technology, we take a designer-centered stance, asking the question: how do professional designers externalize ideas for off-the-desktop computing and how do these inform next generation design tools? We report on interviews with designers from various domains, including experience, interaction, industrial, and space designers. The study broadly reveals perceived challenges of moving into a non-traditional design medium, emphasizes the practice of storytelling for relating the context of interaction, and through two case studies, traces the use of various external representations during the design progression of ubicomp applications. Using paper-prototyped "walkthroughs" centered on two common design representations (storyboards and physical simulations), we formed a deeper understanding of issues influencing tool development. We offer guidelines for builders of future ubicomp tools, especially early-stage conceptual tools for professional designers to prototype applications across multiple sensors, displays, and physical environments.

Keywords

Ubiquitous computing, qualitative study, design practice, external representations, authoring tools, storyboarding

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces – *Prototyping, User-centered Design.*

INTRODUCTION

Weiser's vision for computing predicts applications that span multiple devices, sensors, and physical spaces to become naturally ingrained in how people live [32]. Despite technology under development to infer human activity and

social context from sensors [24], intelligently distribute information across networks of devices, and enable input modalities beyond the keyboard and mouse, we argue that Weiser's vision has not been achieved. Although user-valued applications such as those supporting home elder care [20] and hospital health care [2] have been developed by researchers, we are far from enabling true designers, those who think about *human experience*, to rapidly explore, prototype, and evaluate their ideas using ubicomp technologies.

Many recently-developed authoring tools for ubicomp target specific technologies or are not usable by non-technologists [9,10,12,15,16,17,18]. We want to inform the development of tools for ubicomp by taking a designer-centered approach, looking at external representations and methods used by designers today. Although it is imprudent to claim a design practice exists for ubiquitous computing, our studies reveal instances of off-the-desktop design and show how professional designers cope with the limitations of current communication tools¹. Design tools like Macromedia Director and Flash are inadequate for blueprinting ubicomp systems because these environments only provide support for traditional desktop interaction.

Our field study conducted with eleven designers from various related disciplines reports on useful representations, practices, and tools for ubicomp. Our investigation highlights methods for communicating the interaction context and the bottlenecks that arise. For example, designers often communicate through *contextual storytelling* to engineers who can build a prototype. While storytelling provides a natural vehicle for conveying ideas, it may be problematic for engineers who must translate stories into technical specifications, and for designers who have less control and must spend more time to iterate representations, particularly in contrast to previous design mediums, such as the web. We identify three perceived challenges to creating ubicomp systems from a designers viewpoint: organizational barriers, lack of knowledge, and technical difficulties. We highlight

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, 2006, University Park, Pennsylvania, USA.

Copyright 2006 ACM 1-59593-341-7/06/0006...\$5.00.

¹Since the ubicomp design domain is still emerging, we approximate by looking at experience designers, interaction designers, industrial designers and architects, whose combined design efforts dictate the encounters people have with technology in a physical space.

two common representations for designing ubicomp systems (i.e., contextual storyboards and physical simulations) and illustrate their use with case studies. Based on our findings, we paper prototyped “design tool walkthroughs” to further understand the issues facing tool-users. Finally, we put forth implications that impact the creation of next-generation design tools and briefly describe our future work, including a speculative design for a ubicomp authoring tool.

BACKGROUND

We briefly review existing research on ubicomp design tools and describe our designer-centered approach grounded in qualitative research and design theory. Development APIs, such as the Context Toolkit [10], provide an infrastructure for ubicomp, but are too complicated for most designers. On the other hand, end-user programming and configuration tools for ubicomp, including a CAPella [9] and CAMP [30], are idealized tools incapable of designing the complex ubicomp behaviors desired by professional designers.

There are several ubicomp design environments specifically targeted at designers. DAMASK supports the design of 2D interfaces across multiple devices (PDAs, mobiles phones, etc.) [17]. Topiary allows a designer to specify 2D content changes based on a user’s location context [16]. DART enables media designers to explore augmented reality technology [18]. The Calder Toolkit [12] and d.tools [15] projects are attempts to support interaction and industrial designers to create functional, interactive prototypes of physical devices. Each of these tools are valuable contributions to ubicomp design, but none address a broad notion of ubicomp as systems that may include physical and digital design for multiple devices, broad forms of context sensing, and novel interaction techniques.

Our designer-centered approach for thinking about next-generation design tools is methodologically similar to previous studies of design practice in film, publishing, speech application design, and web design. Katz’s work concentrates on storyboarding and pre-visualization in filmmaking [14], techniques that may potentially be adopted by ubicomp designers. In Sumner and Stolze’s study of speech application designers [28] and Bellotti and Rogers’ study of editorial staff at several publishing companies [4], one commonality is the use of multiple representations during the creation process. Newman *et al*’s study of web designers revealed the importance of paper and sketching of multiple representations during early stage design [22]. Newman and his collaborators revealed three primary representations of a web site (i.e., sitemap, storyboard, and individual pages) and used these as a basis for DENIM, an authoring environment to support early iterative design of web sites. Our qualitative work similarly focuses on identifying external representations used by designers and considers how these could be used as a basis for a design tool.

Norman emphasizes the power of external representations (such as visual aids) or cognitive artifacts in thinking [23]. In Schoen’s theory of design, practitioners solve problems

incrementally by creating explicit design representations that “reflect back” to the designer [26]. Similar to Neuwirth and Kaufer’s analysis of external representations used in the writing process, we hope to parse out why certain representations might be more useful than others in ubicomp design [21]. Yamamoto and Nakakoji’s study of externalization in creative work reflects on how specific representations and interaction design techniques instilled in tools can impact design [31]. Likewise, the concept of design rationale described by Moran and Carroll refers broadly to methods, documentation, and communication of design thinking and can lead to input for creating design tools and for teaching designers [19]. In our study we look intently at representations and drawings created by participants, because they communicate internal concepts and can inform the creation of next generation design tools for ubicomp.

METHODS

Our approach for designing an effective tool is to use qualitative methods, such as those proposed in [5, 13], to shed light on the emerging practice for non-traditional computing design. We conducted interviews with eleven professional designers to learn about current practices and to solicit opinions about a potential design tool. Each interview lasted about 90 minutes and consisted of a demographic questionnaire, a semi-structured discussion about the process of designing and prototyping off-the-desktop applications, and questions relevant to a possible future design tool.

To ground the discussion, we had participants discuss one of their projects with some relation to ubiquitous computing, if one existed. We asked them to describe the stages of design, what was created at each stage, who were the people involved, what tools they used, and what methods were employed. We encouraged participants to sketch design representations on paper and followed up with detailed questions when appropriate. We took notes and collected photos and other artifacts to illustrate the design process they employed. All references to field notes in this paper are marked with the participant number in parentheses.

We ended the interview with questions germane to a future design tool. What are the challenges in designing and prototyping off-the-desktop applications? Where would a new design tool help the most? What representations are useful for communicating about ubiquitous computing?

To provide clarification, to gather more materials on representations in ubicomp design, and to iterate on ideas that were inspired by the first round of interviews, we followed up with two specific designers for a second in-depth 60-minute interview. Finally, in the section on design tool walkthroughs, we present yet another round of interviews with our four interaction designers aimed at gathering data on specific possibilities for authoring.

DEMOGRAPHICS OF DESIGNERS IN THE STUDY

We interviewed eleven professional designers: three experience designers, three interaction designers, two

| Years of experience | # of participants |
|---------------------|-------------------|
| Less than 4 years | 1 |
| 4-8 years | 3 |
| 9 or more years | 7 |

Table 1: Professional experience of designers

architects, two industrial designers, and one graphic designer. We talked to architects and industrial designers because we also wanted to hear perspectives on computing from people who think about the layout, appearance, and human experiences of physical spaces, as well as the objects occupying those spaces. In line with Siedman's guidelines for number of participants, we stopped conducting interviews when we reached a saturation of information [27]. Likewise, the six interaction and experience designers sufficiently represented the primary focus of this study².

The demographic questionnaire revealed commonly used design tools and materials, and helped us understand the designers' experience with computers, programming, new technologies, user evaluation, and the Wizard of Oz³ (WOz) method [8]. Nearly all (ten) of the designers we spoke to had previously designed or played some role in creating non-traditional computing applications, such as contextual photo displays, PDA diet assistants, motion sensor triggered narratives, etc. Only one of the participants exclusively designs ubicomp environments as her primary deliverable; the others only dabbled in ubicomp design.

All but one designer has four or more years of experience in their respective fields (Table 1). About half (five) of the designers worked in design firms with less than 100 people total; the other half worked within design departments for large technology companies. In general, the size of an organization affects the design practice in that corporate design environments appear to have more structured methods and dedicated design roles [22].

Three of the designers had never programmed a computer at all; most stated they occasionally program, generally in Action Script or Lingo, the scripting languages for Macromedia Flash and Director, respectively. Only one designer considered himself an expert programmer in these scripting languages. We presented each designer with a list of technologies such as proximity sensors, RFID (radio frequency identification), home automation devices such as X10, GPS (global positioning system) and other sensors, and asked the designer to rate their familiarity with the technology from a design perspective (rather than from a user's point of view). The majority (seven) of our participants were familiar

² According to the designers in our study, experience designers focus on the emotions, cultural issues, and high-level experiences of people and new technology, where interaction designers think about the visual/tactile details and the lower level interactive aspects of a system.

³ The WOz method is commonly used by designers and HCI researchers to get early feedback from users without major technology development investments by faking some portion of a proposed system.

with how these technologies work and had contemplated their use in projects. However, only three designers had ventured to create working prototypes based on these technologies.

The demographic questionnaire also revealed that because of time and cost, only three of our participants conduct user evaluations on design prototypes. Instead, most use informal methods to evaluate design ideas such as design critiques or grabbing a co-worker to quickly try things out. When user evaluations are performed, they are generally done by a usability expert. Also, all of the designers were familiar with the WOz method for user evaluation, but never used it.

FINDINGS

Not only do the different design disciplines vary greatly in detail, a single designers' process is individually crafted from years of experience. One familiar theme, however, particularly relevant to ubiquitous computing, is the practice of creating stories that explain the context of interaction. When contrasted with previous design processes, such as the tradition for web design, this finding highlights an important practice that could be integrated into a future design tool. Our findings include observations of the design process, perceived challenges by designers for prototyping ubicomp applications, and common design representations used to communicate ideas.

Observations on the Design Process

Across all disciplines, a familiar story emerged about the design process: users' needs are researched, goals are identified, ideas are brainstormed, concepts are iterated and narrowed down, details are added to the chosen concept, and prototypes are created at various scales. Design ideas evolve from conceptual abstractions to more concrete representations. Decisions are balanced with constraints throughout the process. One designer said design always starts on pen and paper—sketching out high-level goals, working out relationships, and often waiting for inspiration (P10, field notes). Although our observations confirmed many of our assumptions about how designers operate, the subtle differences between individuals and between disciplines revealed the richness and eccentricity in design practice.

Pre-Design Research

When preparing for a design project, most designers in our study rely on qualitative methods such as ethnography, focus groups, blog surfing, and informal interviews (to name a few), as well as historical and contemporary references, observations of everyday life, and other resources to decide where to focus their design skills. One designer talked about using all the resources in her life (observations, books, social outings, gossip, politics, etc.) to choose the right concepts and to focus her ideation (P6). Our inquiry mostly focuses on design and prototyping, not the deep user research that often occurs long before design ideas emerge.

Emphasis on Storytelling

A concerted effort must go into conveying the context of human interaction in the proposed world, especially for



Figure 1: An example of storyboarding used to show how new technology fits into a social and physical place.

designs that intersect traditional boundaries. In ubicomp design, due to the technology's novelty and the scope/cost of producing these ideas, **storytelling is crucial**, because ideas either get approval to move into detailed design or they die.

Storytelling takes on different forms from storyboarding (Figure 1) to PowerPoint presentations to video productions to written descriptions of ideas. In the interviews, designers would use words like *experiential*, *perspectives*, *scenarios*, *activities*, etc. to describe the nature of storytelling during early stage design. Building the story around the design ideas helps clarify the issues, particularly for the designer.

Contextual Storytelling Vs. GUI / Web Design Tradition

Storytelling has been used previously by ubicomp HCI researchers to describe a vision of technology. The Starfire video prototype by Sun Microsystems [29] and Bardram et al.'s virtual video prototypes of pervasive health care systems [2] are good examples of the tremendous efforts that often go into conveying a story about a proposed technology.

There is a distinction between storytelling and actually prototyping an idea. With ubicomp applications there is a pronounced need for "contextual" storytelling that speaks to the context of people, physical objects, and information intersecting in time and space—especially in contrast to traditional forms of interface design where the context of interaction can be assumed (i.e., mouse and keyboard desktop interaction in web site design).

Screen storyboards and sitemaps created during early web design are closely analogous to the final medium. Early brainstorming for a web site can be organized into a sitemap by information architects. Screen storyboards show the user's path for a particular task within the web site. Web designers often perform design activities within professional tools that easily transition to web development tools.

The design materials and representations created for setting the context in ubicomp systems share much less with the prototype that follows. There are no good tools to support the range of sensors and displays encompassed in ubiquitous computing. Contextual storytelling is an emerging practice for representing design ideas in ubicomp, but issues exist in translating stories into technical specifications and in providing designers explicit iterative control.

Perceived Challenges for Ubicomp Design

Although ubicomp researchers have recognized challenges for the research community [1, 11], they say little about ubicomp challenges in professional design settings. We identify three main challenges seen by designers to be able to design/prototype within the ubicomp domain: designers face economic and social barriers trying to introduce non-traditional technologies to clients and potential users; designers do not have enough information about new technologies to understand the feasibility constraints, physical properties, and the vectors of change; and designers find it difficult to prototype ubicomp applications due to the level of complication, lack of tool support, and the unfettered, unstable nature of new technologies.

Organizational Barriers

Nearly every designer talked about convincing others (clients, higher execs, engineers, others on their own team, etc.) about the benefits of their ideas, which speaks to the importance of storytelling. One participant said, "it's about making arguments to clients... about selling the fantasy of how people will use the space" (P3). Financially speaking, a design project will not move forward unless clients or executives understand the value of proposed ideas.

The social settings for design, particularly in corporations, can be challenging with respect to gaining acceptance and financial backing for a design idea as well as in coordinating the design effort among a team of people with different strengths. Several participants expressed the distributed nature of design, where one's role is dictated by specialized skills and organizational structures (Figure 2). For example, one designer might design the user experience and then pass it to specialists who design the visual, physical, and information flow details. The designers generally work closely with engineers to understand technical constraints and to communicate specifications for what should be built. Design tools should take into account the social nature of design and provide a communication mechanism for different design roles.

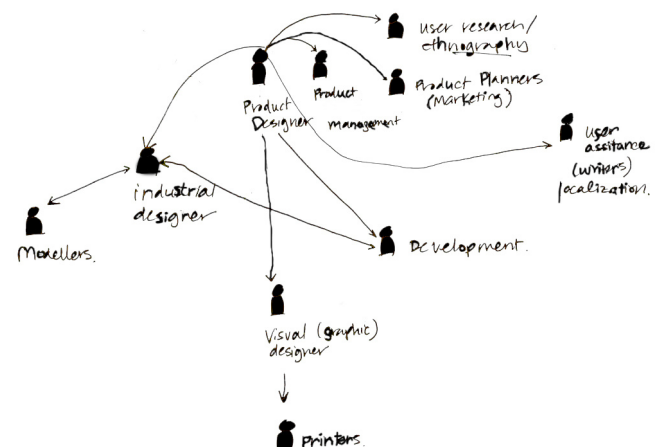


Figure 2: One designer's sketch of the distributed nature of design within a corporate environment (P1).

Lack of Knowledge

Designers often do not have the right information to understand the role a particular technology could play in application ideas. Nor do they have information about feasibility constraints and physical properties that impact design (for example, understanding the range for RFID tag detection). One designer explained that the knowledge rests in many different heads and that it's too hard to know about the capabilities and trade-offs of new technologies (P5).

The architects we spoke to raised interesting issues about installation and longer-term infrastructure issues, confirming previous research observations by Rodden et al [25]. Architects already design around ubicomp technology because they deal with smart security systems, temperature controls, and so on. They view ubicomp technologies as utilities, and often rely on experts to help them determine design trade-offs. An important issue for architects is whether or not an infrastructure installed today will be obsolete in a few years. One architect pointed out how technology changes much more rapidly than plumbing systems, so they are very cautious about installing expensive infrastructure (P10). They need to have the right knowledge about the startup and long-term costs of installation, and to be able to effectively communicate why it's important to clients. A challenge for design tools is conveying the ever-evolving state of technology as well as the current constraints and possibilities.

Technical Difficulties

As one designer states, they would happily jump right into a real technology prototype if it were cheap, reliable, and required no maintenance—unfortunately, this is rarely the case (P9). According to another designer, a physical mock-up of an idea, although costly, is especially effective for communicating to executives, clients, other researchers, users, and ourselves (P5). These statements motivate the need for a prototyping tool, but highlight an issue that will continue to be a problem even with tool support. Issues of reliability, cost, maintenance, etc. are engineered into the technology, so providing an authoring layer will not prevent these problems and may inappropriately conceal the issues. Design tools should be tolerant and flexible, potentially allowing virtual proxies for real devices that fail.

Representations in Ubicomp Design

During the interviews we asked designers to sketch or talk about high-level representations or useful metaphors for expressing application ideas. The most common responses provided were physical portrayals (such as a map or a system diagram), stories about the experience of users, and abstract node diagrams expressing information flow. In additional interviews with two designers, we observed the first two of these metaphors through the materials and representations used in practice. We present two case studies of design that elucidate the creation of storyboards and the use of maps.

Case Study 1: Storyboard Representation

In this case study, a team of designers in a medium-sized design firm did preliminary designs on a new physical

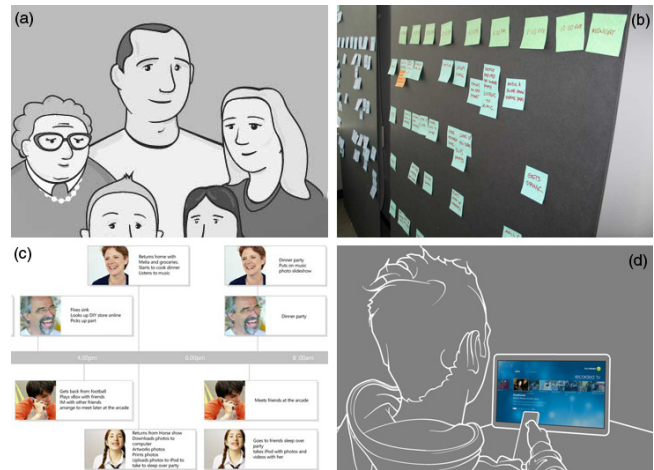


Figure 3: Constructing a design story: a) creating personas for potential users of the new technology b) brainstorming activities for the fictitious family throughout a day c) visualizing a script that details the family's interaction with technology over time d) storyboarding the context of interaction and user experience to communicate with clients, other designers, and engineers.

computing device and its software. They started by amassing their preliminary research into the creation of personas (Figure 3a). They created a fictitious family based on user research, complete with names, personalities, and sketches. Next, within the design firm they held brainstorming sessions and charted out the possible activities for the entire family throughout a day, including their interaction with the new product. Figure 3b shows the outcome of the “day in the life” brainstorming, done on a large board with post-it notes. The designer we spoke to described this as a late night brain dump with lots of people collaborating (P11).

At this point, the timeline of activities on post-it notes were put together into a more formal presentation for the client (Figure 3c). The best ideas were narrowed down into an organized narrative that traces the family and their interaction with technology throughout a day. After getting feedback on the story from the client, they did another story iteration to make it more concrete. They created a formal script, and started working on a photographic storyboard. To create their storyboard, they gathered people from around the office to pose in different physical positions (sitting at a table, standing with headphones, etc.) The photos provided a realistic outline of people; using Photoshop they augmented the photos with technology sketches to create nice looking storyboards. The product of this round of storytelling was a series of storyboard cells presented in an interactive Flash demo (Figure 3d). At the time of the interview, the design team had just presented the storyboard demo to the client and was waiting to learn if the project would move forward.

Case Study 2: Physical Representation

The second case study follows the design trail of a demo idea for a large tech company's futuristic home, told from the perspective of the home's lead designer. Here we essentially pick up where the first case study ended, with comic book



Figure 4: This designer would augment a map with numbers to organize several stories or themes. Our participant sketched the map; we overlaid the storyboard images provided by the designer to show what the numbers refer to conceptually.

style storyboards showing the interaction of people with a proposed technology. The process to reach those storyboards was similar to the process in the first case. From here, the designer sketched on paper how she would communicate with other designers and engineers over a map of the home. Since they were designing several technologies, each with an associated story, they used the map to organize the narratives into one representation (Figure 4).

The purpose of the futuristic home is to show off new technology to visitors, so the map became a natural way to work out the flow of the visitor experience. According to the designer, the storyboards and map plus a three page written design specification are the final deliverables to the engineers who will build the technology. The engineering team would often divide the storyboards into independent projects for each engineer. Having only the high-level storyboard and the written specification, the engineer needs to be in close communication with the designers to clarify details.

The case studies illustrate the role of storyboards and physical portrayals in painting the context of interaction for ubicomp technologies. There is a big difference between storyboarding and prototyping, often resulting in a rift between designers and engineers, where engineers must interpret contextual storyboards and fill in the design details. Additional tool support will help designers and engineers communicate users needs and technical constraints.

DESIGN TOOL WALKTHROUGHS

Our initial interviews and observations motivated the need for explicit authoring support, but we had deeper questions about how perceived challenges and current external representations might impact the design of a tool. How much explicit support should be offered for transitioning between contextual storytelling materials and a concrete form that can be interpreted by a computer as a prototype? What would it mean to structure a ubicomp design tool around metaphors

such as storyboards or physical maps? Which one is more suited as a tool metaphor and how will it impact the creative work?

In this section we describe “design tool walkthroughs”, or environments prototyped on paper used to provide deeper insight on implications for design tools for multi-sensor, multi-device ubiquitous computing. We conducted another set of interviews with four of the interaction/experience designers from the previous study. In the interview we walked through two paper design environments, one based on the storyboard metaphor and one based on a physical simulation metaphor, and discussed how each could be used to design specific ubicomp applications. We asked the designers to describe advantages and disadvantages of each authoring paradigm and to reflect on representations and tool specifics most useful for design.

Description of Paper Design Tools

Guided by our research questions regarding integration of storyboarding practice, as well as the concepts expressed by designers and reflected in materials we accumulated during the initial interviews, we conceived of and paper prototyped two alternate design tools based on two different metaphors: a “storyboard” metaphor that takes advantage of the existing practice of contextual storyboarding (using sketches, photos, videos, etc. to tell a story) and a “physical simulation” metaphor based around a designer’s representation and simulation of physical space and the events that occur over time in the space. These were not meant to be final designs of tools, but rather useful research probes.

Storyboard Metaphor

With the storyboard-based metaphor, our simulated tool allows you to sketch or import a contextual storyboard (Figure 5a). The designer can group storyboard cells and then convey how technology will interact by placing “building blocks” into the appropriate groups. The building blocks are designer defined placeholders that succinctly describe an event or a relationship between input sensors (“conditions”) and output displays (“responses”). Designers can later associate actual devices with each condition and response, and then use a recording mechanism to set properties on devices. For example, designers might create a building block with a condition called “book near light” and a response called “light on”. The “light on” response is later formalized for the computer by selecting that particular building block, hitting record, selecting a lamp from a list of known devices, and setting the lamp’s power property value to *on*.

Physical Simulation Metaphor

In the physical simulation-based metaphor, the designer sketches or imports an image that represents the target environment(s). This could be a 3D model similar to Barton’s work on the UbiWise simulator [3], an overhead map of one or multiple places, or a system architecture showing all the possible devices and sensors for the application.

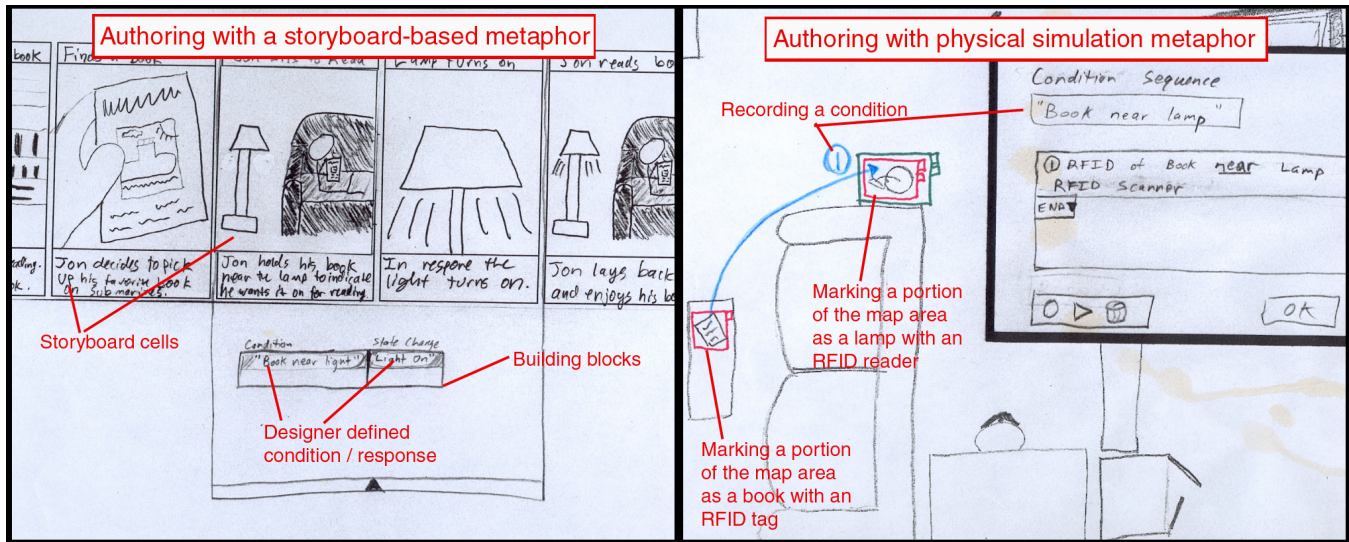


Figure 5: a) Paper prototype of a design tool based on the storyboard metaphor. Designers give details underneath story cells (condition “book near light” causes the response “light on”). b) Design tool based on a physical simulation of things in the real world. Designers situate and record sensors and displays in particular states to design the conditions and responses within the space.

As an example, a designer can draw a picture of a lamp and a book (Figure 5b), and then “mark” or highlight the respective portion of the image as an actual device. So, the sketched lamp in the image represents an actual lamp with physical properties. The map becomes a simulation of the actual physical world, and can potentially be used to emulate and visualize properties such as ranges for radio-based devices, form factors for physical devices, and the appearance of various displays. Conditions and responses are similarly recorded by interacting with the “marked” portions of the image—moving devices around to cause proximal conditions and accessing their properties to express conditions for many types of sensors. When a designer runs the application they manipulate objects directly in the physical simulation, which visualizes how events play out and the state of devices.

Obtaining Feedback from the Design Tool Walkthroughs

We conducted design tool walkthroughs with four designers from the first study. Since our simulated tools emphasize the design of complex interactions of people and information across multiple sensors, displays, and environments, we only spoke with experience and interaction designers most familiar with ubiquitous computing. The feedback sessions lasted 90 minutes and included a high-level explanation of each tool metaphor and example applications, followed by a walkthrough of the details in each paper prototype. The example ubicomp applications were necessary to explain how each tool could work under a particular metaphor. When we felt each participant understood each tool idea, we asked them to tell us what they perceived to be the advantages and disadvantages of each paradigm.

Throughout the session, we clarified all the language used in the prototypes and reached a common understanding with each participant before moving forward. At the end of the session in an attempt to better align our use of language

with the designers understanding, we asked the designers to choose from a list of alternative phrases (for example, instead of using “state change” to refer to output on devices, most of the designers we talked to preferred “response”).

Advantages/Disadvantages of Each Tool Metaphor

We solicited feedback about the advantages and disadvantages of each interface metaphor, and asked detailed questions about our design decisions. According to our participants, the advantages of using a storyboard-based metaphor include the ability to express the intimacy and emotion of human relationships, the potential to utilize pre-created storyboards, the support for designing time-based interactions, the ability to work top-down with designer-defined language, and the modularity achieved with the event building blocks. The disadvantages of the storyboard interface are that it forces the designer to create storyboard sketches for portions of the application that do not have a compelling story or normally wouldn’t have an associated storyboard, and that it does not allow you to see all the events at one time. Two designers noted the risk of missing logical connections, getting trapped within the story, and not being able to realize feasibility issues with their design.

The design tool based on physical simulation avoids some of these pitfalls, providing an overview of the entire system and allowing designers to better recognize issues and capabilities. It provides for more natural and flexible user interaction, because of the support for sketching, marking portions of the image as devices, and gesturing to record conditions. Our participants pointed out several disadvantages of this interface metaphor including the difficulty of representing time and the potential for overly complex and cluttered physical simulations. It could also slow down expert designers because it requires the designer to sketch everything and then define the interaction.

In general, the designers understood the value of having a design tool; the tool could be an extension of their storytelling ability and a more detailed way to create specifications for technologists (P9). Designers hope a future tool provides a better indication of technological constraints, as seen in statements like “Storytelling is often utterly unrealistic; this tool should bring us back to earth” (P6). Designers also expressed desire for a realistic emulation of physical and social properties in the world, so called “real world logic” (P5). The reactions to our walkthroughs provided deeper insight into implications for future design environments.

GUIDELINES FOR UBICOMP TOOL-BUILDERS

Through initial interviews and “design tool walkthroughs”, the designers we spoke with expressed many useful ideas towards the design of an authoring environment for ubicomp. Many designers brought up established design doctrine, such as iteration, layering, and expressiveness, which should be instilled in any design tool. The considerations listed here speak to the builders of next generation design tools and aspire to guide the development of research and commercial tools for professional designers. Next generation design environments for ubicomp applications should provide:

- Multiple representations
- Designer-defined events
- Knowledge support
- Real and fake device support
- Simulation modes
- Integration with the social context of design
- Accordance with other design tools

Multiple Representations

In ubicomp design, the designer defines the cause/effect relationships that occur among sensors and displays interwoven with human activities; providing multiple representations of those relationships allows designers to choose an expressive mode most appropriate for the current stage of design or a particular design perspective. Our participants found advantages and disadvantages of both the physical simulation and storyboarding metaphors, and most suggested a combination of representations, perhaps something that better summarizes events and information flow, especially for expert designers. Ideally, changes made in one representation should be reflected in the other views so that a designer can move between them fluidly utilizing the best representation for the design task at hand.

A natural evolution between representations may emerge in the ubicomp design process. For example, a designer might start with a storyboard representation to explain high-level concepts. After several stories describing relationships between sensors and displays within an environment, the designer might move to a physical simulation view to pull together disparate stories and technologies into an overview. Yet another representation, such as an information/event view, might be used for detailed logic design such as conditionals, sensor thresholds, and device settings.

Designer-Defined Events

One feature particularly favored by participants in the paper prototypes was the ability to use any written language to define cause/effect relationships. This event model provides a linking mechanism between the conditions of sensors in the world and the outcomes designers want in response. Although these logical links can be made in other ways, such as using a wiring metaphor like Max/MSP [7], designer defined language allows designers to “work through” the problems over time, adjusting labels and elaborating as appropriate. This is especially important in ubicomp where designers are already overloaded with device jargon and there is no prevailing interaction model, as is the case with desktop interaction.

Knowledge Support

One challenge expressed by designers in the interviews and the walkthroughs is the need for more knowledge about available and soon-to-be available technologies. Next generation design tools should build in support for learning and knowledge propagation. For example, a tool might provide a device catalog that represents available devices, and how designers might use them in an application. Akin to the design patterns work [6], the catalog could give a textual description of common uses, constraints, opportunities, and other relevant information. For example, it might be pertinent to know that GPS devices perform poorly indoors or near dense, tall structures. A design tool could also go as far as suggesting available and appropriate technology by parsing out designer-defined language used in setting up high-level event relationships and checking against existing knowledge sources.

Real and Fake Device Support

The ability to support real and fake devices is important for designers because it supports a naturally iterative process. Initially fake devices can be used to emulate a ubicomp system, providing a compelling demonstration in it’s own right. Assuming modularity between real and fake devices, real devices can be plugged-in to replace emulators as they become available over time. If a ubicomp system is composed of entirely real devices, fake devices can stand-in when a component fails, sustaining robustness for the rest of the prototype. This guideline speaks to the unreliability of ubicomp technologies and to designers’ natural process by allowing designers to see the outcome of their designs without creating a full implementation.

Simulation Modes

A design tool should allow a designer to continually test applications by entering a simulation or “run” mode. If real devices are connected, the designer can test the responses of the devices in the actual space. If real devices have not been connected and the designer is using emulators, the simulation should understand how the system should behave under certain conditions. As one participant suggested, it would be interesting to test a design under certain social and

physical simulations to see how it “plays out”. At a minimum a designer should be able to manually interact with fake devices to trigger a particular device condition or event to understand the “ripple” effect in the rest of the application.

Integration with the Social Context of Design

Ubiomp design will likely be a collaborative effort involving multiple disciplines and special skills. Similar to other massive design efforts such as filming movies, large-scale ubiomp applications will likely require a division of labor among a crew with dedicated roles. Design tools should take into account the social nature of design and provide a communication mechanism for different design roles. Potentially, a formal specification can be generated for engineers who might take the design prototype and build a robust, tightly engineered system that will be placed in the real world. Design does not happen in a vacuum with one designer tinkering in a problem space. It is a social effort between designers, users, clients, engineers, and other stakeholders.

Accordance with Other Design Tools

Similarly, a ubiomp design tool should work together with other tools. It should aggregate content from specialty tools like image manipulators, 3D modelers, sound editors, storyboard tools, simulation tools, video production tools, and so on. For example, many inventions in ubiomp will involve new physical forms for devices, so the tools should be able to import 3D models created by the industrial designers as representations for fake devices in a physical simulation. The design tool should also create simple media products, such as a Quicktime movie or a downloadable plugin for the browser, to share with collaborators over a distance who may not have the full design environment.

SPECULATIVE NEXT GENERATION DESIGN TOOL

To provide inspiration for tool designers, and to give concrete form to the guidelines laid out in the previous section, we have designed a speculative “next generation” design tool for ubiomp (Figure 6). Compelled by the practice witnessed in the second case study, we created a physical simulation area for the design tool to serve as the hub for multiple storyboard representations. A third representation, the event overview, exhibits every condition and response created through either the map or storyboard view, providing an important summary of system relationships and a shortcut for experts who can cut/paste and tweak details without creating storyboards or sketches. In our design, changes made in one view will be automatically reflected in the others.

Our team is currently engineering a working authoring environment using Universal Plug and Play (UPnP) and Java as an underlying infrastructure for connecting sensors and displays. The next step is to build the user interface for the tool and connect it to the device infrastructure. We expect to work with designers as we build the tool and to continually evolve our ideas. Eventually, we will recruit professional designers to create applications, performing both specific and open-ended tasks, to formally evaluate the design tool.

CONCLUSION

Although many researchers are working on design environments related to ubiomp technologies, no one is investigating existing design practice and tools to orient future tool research. Our approach was to conduct qualitative studies with professional designers through interviews and “design tool walkthroughs” to discover perceived challenges and common representations for designing ubiomp applications. The study led to a list of design considerations aimed at tool-builders. We concluded that next generation

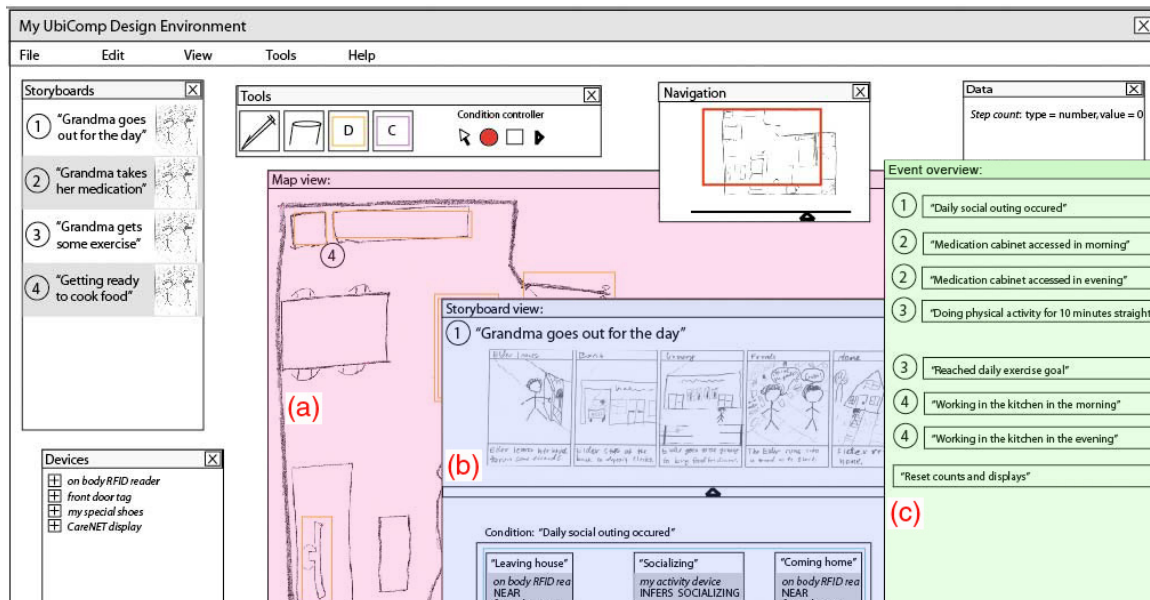


Figure 6: Sketch of a speculative Ubiomp design environment. Three different views (highlighted with color) provide information about the event relationships: (a) map view shows a physical simulation of the space, (b) storyboard view shows events associated with a particular high-level story, and (c) event overview shows every relationship defined in either the storyboard or map view.

design environments for ubicomp applications should provide multiple representations, designer-defined events, knowledge support, real and fake device support, simulation modes, integration with the social context of design, and accordance with other design tools.

We hope these guidelines will inform ubicomp tool builders and lead to designer-centered authoring environments to support rapid, exploratory design and early prototyping of multi-sensor, multi-display systems by non-programming, professional designers and creative thinkers.

ACKNOWLEDGEMENTS

We want to acknowledge Wendy March, Flora Goldthwaite, Drew Bamford, Samuel Hoang, Amy Volda, Gregory Abowd, Andrea Forte, Scott Klemmer, Sunny Consolvo, Gillian Hayes and Khai Truong for their various contributions and thanks to Intel Research Seattle for hosting the project.

REFERENCES

1. Abowd, G.D. and Mynatt, E.D. Charting Past, Present and Future Research in Ubiquitous Computing. *ACM Transactions on Computer-Human Interaction, Special issue on HCI in the new Millennium*, 7(1):29-58, March 2000.
2. Bardram, J. E., Bossen, C., Lykke-Olesen, A., Madsen, K. H., and Nielsen, R. Virtual Video Prototyping of Pervasive Healthcare Systems. In *Proc. of Conf. on Designing Interactive Systems (DIS'02)*, 2002, 167-177.
3. Barton, J. and Vijayaraghavan, V. UBIWISE: A Simulator for Ubiquitous Computing Systems Design. *Tech. Report HPL-2003-93, Hewlett-Packard Labs*, 2003.
4. Bellotti, V. and Rogers, Y. From Web Press to Web Pressure: Multimedia Representations and Multimedia Publishing. In *ACM Proc. of Human Factors in Computing Systems (CHI '97)*, 1997, 279-286.
5. Blomberg, J., Giacomi, J., Mosher, A., and Swenton-Wall, P. Ethnographic Field Methods and Their Relation to Design. In *Participatory Design: Principles and Practices*. Schuter, D. and Namioka, A., Editors. Lawrence Erlbaum Assoc, 1993, 123-155.
6. Chung, E.S., Hong, J.I. Lin, J. Prabaker, M.K. Landay, J.A., and Liu, A. Development and Evaluation of Emerging Design Patterns for Ubiquitous Computing. In *Proc. of Designing Interactive Systems (DIS2004)*, 2004, 233-242.
7. Cycling 74's Max/MSP, <http://www.cycling74.com/>
8. Dahlback, N., Jonsson, A., and Lars, A. Wizard of Oz Studies: Why and How. In *International Workshop on Intelligent User Interfaces (IUI'93)*, 1993, 193-200.
9. Dey, A.K., Hamid, R., Beckmann, C., Li, I., and Hsu, D. a CAPpella: Programming by Demonstration of Context-Aware Applications. In *ACM Proc. of Human Factors in Computing Systems (CHI '04)*, 2004, 33-40.
10. Dey, A.K., Salber, D., and Abowd, G.D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, 16(2-4), 2001, 97-166.
11. Edwards, W.K. and Grinter, R. At Home with Ubiquitous Computing: Seven Challenges. In *Proceedings of the Conference on Ubiquitous Computing (ubicomp 2001)*, 2001, 256-272.
12. Hartmann, B., Klemmer, S.R, Bernstein, M., and Mehta, N. d.tools: Visually Prototyping Physical UIs through Statecharts. In *Ext. Abstracts of ACM Symposium on User Interface Software and Technology (UIST 2005)*, 2005.
13. Hughes, J., King, V., Rodden, T., and Andersen, H. The Role of Ethnography in Interactive Systems Design. *Interactions*, vol. 2(2), 1995, 56-65.
14. Katz, S. *Film Directing Shot by Shot: Visualizing from Concept to Screen*. Michael Wiese Productions, Studio City, CA, 1991.
15. Lee, J.C., Avrahami, D., Hudson, S., Forlizzi, J., Dietz, P., and Leigh, D. The Calder Toolkit: Wired and Wireless Components for Rapidly Prototyping Interactive Devices. In *Proc. of Designing Interactive Systems (DIS2004)*, 2004, 167-175.
16. Li, Y., Hong, J.I., and Landay J.A. Topiary: A Tool for Prototyping Location-Enhanced Applications. In *ACM Symp. on User Interface Software and Technology (UIST'04)*, 2004, 217-226.
17. Lin, J. and Landay, J.A. Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. In *Proc. of Conf. on Distributed Multimedia Systems (Intl. Workshop on Visual Computing)*, 2002, 573-580.
18. MacIntyre, B., Gandy, M., Dow, S., and Bolter, J.D. DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences. In *ACM Symp. on User Interface Software and Technology (UIST'04)*, 2004, 197-206.
19. Moran, T.P. and Carroll, J.M. *Design Rationale: Concepts, Techniques, and Use*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1996.
20. Mynatt, E.D., Rowan, J., Jacobs, A. and Craighill, S. Digital Family Portraits: Supporting Peace of Mind for Extended Family Members. In *ACM Proc. of Human Factors in Computing Systems (CHI '01)*, 2001, 333-340.
21. Neuwirth, C.M. and Kaufer, D.S. The Role of External Representation in the Writing Process: Implications for the Design of Hypertext-Based Writing Tools. In *Proc. Hypertext and Hypermedia*, 1989, 319-341.
22. Newman, M.W., Lin, J., Hong, J.I., and Landay, J.A. DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice. *Human-Computer Interaction*, 18(3), 2003, 259-324.
23. Norman, D.A. *Things That Make US Smart*. Addison-Wesley, New York, 1993.
24. Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox D., Kautz H., and Hähnel, D. Inferring Activities from Interactions with Objects. *IEEE Pervasive Computing*, October, 2004, 50-57.
25. Rodden, T. and Benford, S. The Evolution of Buildings and Implications for the Design of Ubiquitous Domestic Environments. In *ACM Proc. of Human Factors in Computing Systems (CHI '03)*, 2003, 9-16.
26. Schoen, D. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York, 1983.
27. Seidman, I. *Interviewing as Qualitative Research*, Teachers College Press, New York, 1998, p 47-48.
28. Sumner, T. and Stolze, M. Evolution, Not Revolution: Participatory Design in the Toolbelt Era. In *Computers and Design in Context*, Kyng, M. and Mathiassen, L., Editors. MIT Press, 1997, 1-26.
29. Tognazzini, B. Starfire: A Vision of Future Computing (by Sun Microsystems). www.asktog.com/starfire/starfireHome.html.
30. Truong, K.N, Huang, E. M., and Abowd, G.A. CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home. In *Proc. Of Ubiquitous Computing (ubicomp'04)*, 2004, 143-160.
31. Yamamoto, Y. and Nakakoji, K. Interaction Design of Tools for Fostering Creativity in the Early Stages of Information Design. *International Journal of Human-Computer Studies*, (63) 2005, 513-535.
32. Weiser, M. The Computer for the Twenty-First Century. *Scientific American*, September, 1991, 94-100.