

Development and Evaluation of Emerging Design Patterns for Ubiquitous Computing

Eric S. Chung¹, Jason I. Hong¹, James Lin¹, Madhu K. Prabaker¹, James A. Landay², Alan L. Liu²

¹Group for User Interface Research

Computer Science Division

University of California

Berkeley, CA 94720 USA

{jasonh, jimlin}@cs.berkeley.edu

²DUB Group

Computer Science and Engineering

University of Washington

Seattle, WA 98195 USA

landay@cs.washington.edu

ABSTRACT

Design patterns are a format for capturing and sharing design knowledge. In this paper, we look at a new domain for design patterns, namely ubiquitous computing. The overall goal of this work is to aid practice by speeding up the diffusion of new interaction techniques and evaluation results from researchers, presenting the information in a form more usable to practicing designers. Towards this end, we have developed an initial and emerging pattern language for ubiquitous computing, consisting of 45 pre-patterns describing application genres, physical-virtual spaces, interaction and systems techniques for managing privacy, and techniques for fluid interactions. We evaluated the effectiveness of our pre-patterns with 16 pairs of designers in helping them design location-enhanced applications. We observed that our pre-patterns helped new and experienced designers unfamiliar with ubiquitous computing in generating and communicating ideas, and in avoiding design problems early in the design process.

Author Keywords

Design, Design patterns, Ubiquitous computing

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces—Theory and methods, Style guides, Evaluation/methodology

INTRODUCTION

Design patterns have been proposed in many domains as a format for capturing and sharing design knowledge between practitioners (e.g., [2-5, 9, 21, 23]). Patterns communicate insights into design problems, capturing the essence of recurring problems and their solutions in a compact form. They describe the problem in depth, the rationale for the solution, how to apply the solution, and some of the trade-offs in applying the solution. A set of interlinked patterns for a specific domain is known as a *pattern language*.

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.

DIS2004, August 1-4, 2004, Cambridge, Massachusetts, USA.
Copyright 2004 ACM 1-58113-787-7/04/0008...\$5.00.

Patterns differ from other formats for capturing design knowledge, such as guidelines and heuristics, in three ways. First, patterns offer solutions to specific problems rather than providing high-level and sometimes abstract suggestions. Second, patterns are generative, helping designers create new solutions by showing many examples of actual designs. Third, patterns are linked to one another hierarchically, helping designers address high-level problems as well as low-level ones. Patterns are not intended to replace guidelines and heuristics but rather complement them. Patterns are simply another tool for helping designers create high-quality solutions.

Pattern languages started in the field of architecture [2], and have been emerging for UI design (e.g., [4, 7, 19, 22]) as well as for web design (e.g., [10, 21, 22]). Patterns have seen their greatest success in the area of software design as exhibited by the success of the Gang of Four book *Design Patterns* [9], as well as by the widespread usage of their pattern names within the software development community. This last point represents another important contribution of design patterns, which is providing a common, shared vocabulary that lets designers communicate more easily.

Here, we extend on the idea [12] of using design patterns as a format for assisting designers developing applications for *ubiquitous computing* (ubicom), systems that make use of sensors, computing devices in a variety of form factors, and wireless networking to assist us in all kinds of tasks [24].

Although ubicom is still in its nascent stages, there are many potential benefits in developing a pattern language now. First, we can speed up the diffusion of new interaction techniques and evaluation results by presenting it in a form more usable to designers. Second, a pattern language for ubicom can help us more clearly see links between ideas, as well as what issues remain to be addressed. Third, we can positively influence the design of emerging applications by helping designers find good solutions and avoid adopting poor standards, such as inadequate privacy protection and blue web links¹. As an analogy, when the

¹ As noted by several designers (e.g., [14, 18]), blue is one of the worst colors for unvisited links because of the structure of the human eye. However, since so many web pages use blue for unvisited links and so many people have learned this meaning, blue links have become a de facto standard.

periodic table was initially developed, Mendeleev did not force the known elements to fit in. Instead, he left holes that helped others make predictions about unknown elements that eventually led to their discovery.

Towards this end, we introduce an initial pattern language for ubicomp. This language has 45 pre-patterns addressing application genres, physical-virtual spaces, interaction and systems techniques for managing privacy, and techniques for fluid interactions. We call these *pre-patterns* because they are still emerging and are not in common use yet by the design community and end-users. However, we have found the format of patterns to be useful in communicating design knowledge, even if that knowledge is not set in stone. We expect the pre-patterns to evolve over time, with some being replaced by new patterns.

We have also conducted what is, to the best of our knowledge, the first controlled study of patterns with designers. In our first round of evaluation, we had nine pairs of designers create an initial design for a location-enhanced application, four of which had access to our pre-patterns and five that did not. In our second round, we modified the patterns based on feedback in the first round, and had six pairs of designers use our patterns and one not. This let us compare six pairs of designers in each of the two conditions in the second round.

In the first evaluation round, there were no statistically significant differences in quality, completeness, or creativity between the designs of pairs that used patterns and pairs that did not. In the second round, there were some statistically significant differences with respect to factors such as accomplishing tasks more quickly and usefulness, although most of the differences were between expert and novice designers, rather than between pairs that used patterns and those that did not. However, our qualitative observations in both rounds suggest that patterns helped novice designers generate designs, helped experienced designers new to ubicomp learn about the domain, helped designers communicate ideas, and helped designers avoid potential design problems earlier in the design process. Surprisingly, although we had an entire group of patterns devoted to privacy, our patterns did not help with that issue. Generally, designers found our pre-patterns useful.

RELATED WORK

Design patterns were first developed by Christopher Alexander and his colleagues [2]. Alexander believed that patterns could empower both architects and their clients by providing a living and shared language for design. He and his colleagues developed 253 patterns for building and planning towns, neighborhoods, houses, gardens, and rooms. The emphasis here was on an entire *language* for design, since the usefulness of patterns was not only in providing solutions to common problems, but also in seeing how they intertwined and affected one another.

Our pattern language for ubiquitous computing uses the definition generated at INTERACT '99: "The goals of an HCI

pattern language are to share successful HCI design solutions among HCI professionals..." [4]. An alternative and somewhat complementary perspective is to have a pattern language that is a "lingua franca" for all design stakeholders [6, 8]. While this latter approach has potential for participatory design, it is not one we focused on in the development and evaluation of our design patterns.

Typically, patterns are evaluated through peer review, often in pattern writing workshops [1, 13]. Although the idea of design patterns has been around for quite a while, only recently has there been work on evaluating patterns. For example, Borchers developed and evaluated a pattern language for interactive music exhibits, by having the patterns peer-reviewed at a pattern workshop, by using those patterns in developing two interactive music exhibits, and by surveying undergraduate students that had used some patterns for usefulness and memorability [4]. Dearden et al evaluated a set of patterns for developing airline and rail travel sites with potential users of the system rather than with designers. They investigated whether the patterns empowered users to participate in participatory design and could help users generate designs [6].

However, to the best of our knowledge, there has not been a study evaluating the effect of patterns on how designers communicate and design with one another. We take a first step towards this, looking at how several pairs of designers used our patterns, in terms of learning about a new domain, communicating with one another, evaluating existing designs, and generating designs.

A LANGUAGE OF PRE-PATTERNS FOR UBICOMP

In this section, we give an overview of the method we used to create our pattern language for ubiquitous computing, as well as a description of the pre-patterns themselves.

Developing the Language

Developing the pattern language was an iterative process lasting several months. We started by brainstorming pattern candidates based on a review of the existing literature. We initially tried to create high-level patterns, ones that are fairly abstract and describe whole applications, as opposed to single screens, for example. This proved to be difficult to do, as it meant trying to create a hierarchy of patterns as well as the patterns themselves at the same time. It turned out to be far easier to work bottom-up, identifying relatively low-level and medium-level individual patterns, and then later drawing themes from those to connect them together.

We generated rough cuts of about 80 fairly broad pattern candidates, which looked at a range of interaction and infrastructural issues in ubicomp. There was also a strong focus on patterns for location-based computing, since a sizeable number of this type of application are emerging in the market, and thus has a clearer path to widespread deployment than other areas of ubiquitous computing [15]. We later dropped the infrastructural patterns, instead concentrating on interaction issues because we wanted to focus on helping interaction designers.

Some of these pre-patterns dealt with high-level fundamental issues cutting across all areas of ubicomp (e.g., privacy), while others were relatively low-level interaction techniques (e.g., pick and drop [16]). We tried to find at least two examples for each pattern, and generally preferred commercial implementations since that indicated that the pattern was more likely to find widespread usage.

We then did a card sort [20] among ourselves to organize the pattern candidates into different pattern groups, where each group is a set of patterns with a common theme. After this, we created the main content for each pattern, describing the problem and solution, providing several examples, and establishing links to related patterns. Each pattern was written and edited by at least three of the authors, and was limited to two pages to make it more digestible for the designer. We removed weaker pattern candidates and added new ones where it made sense.

We then solicited feedback from four other researchers familiar with ubiquitous computing. The researchers were first given the name of a pattern and asked to guess what kind of content the pattern would contain. Then, they were shown the full pattern, and asked to rate the quality of the pattern name and to comment on the actual content. We revised the patterns based on this feedback.

At the end, we had 45 patterns in four groups (see Figure 1 at the end of this paper):

- *Ubiquitous Computing Genres* describes broad classes of ubicomp applications.
- *Physical-Virtual Spaces* looks at how physical objects and spaces can be merged with the virtual.
- *Developing Successful Privacy* describes policies and mechanisms for managing end-user privacy.
- *Designing Fluid Interactions* details interaction techniques with sensors and devices.

All of the pre-patterns used for both rounds of evaluations are at <http://guir.berkeley.edu/projects/patterns>

Format of Patterns

The format of our patterns is similar to those in *The Design of Sites* [21] and *A Pattern Language* [2]. Figures 2 and 3 (located at end of this paper) show the first page of several of our pre-patterns. Each pre-pattern consists of:

- A *name* and a *letter-number* pair, where the letter indicates which group the pattern belongs to. For example, “A3” means the third pattern in pattern group A – Ubiquitous Computing Genres.
- The pattern’s *background*, which provides the context and scope of the pattern, and describes any other patterns that lead to this pattern.
- The *problem* that the pattern is addressing.
- The *solution* (or solutions) to the pattern’s problem, as well as pointers to other lower-level patterns that help solve the problem.
- *References* of work related to the pattern.

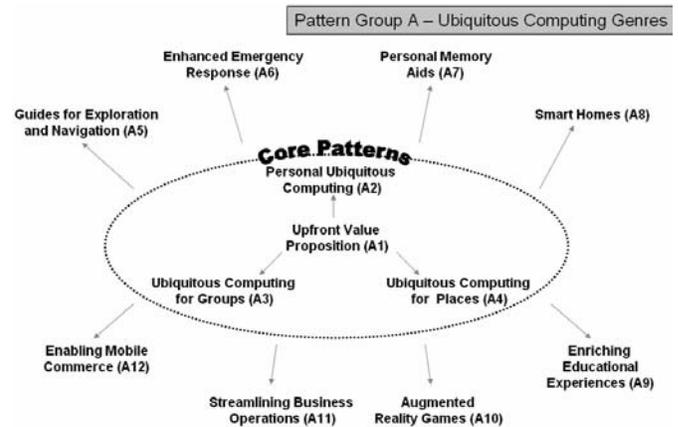


Figure 4. “Bus maps” show how patterns within a pattern group are related. Here, there are four core patterns that are fundamental in designing applications (A1 through A4), as well as several patterns specific to application genres.

We also created what we call “bus maps.” Each map shows the core patterns in a pattern group and the relationship of the patterns within that group (see Figure 4).

Our patterns tended to be more prescriptive than descriptive, mostly because there are few ubicomp applications in practice. Our patterns also tended to focus on high-level issues, such as user needs, versus specific user interfaces and interaction techniques. This is because many of these high-level issues are better understood than the low-level techniques for implementing them. For example, many people have outlined what needs smart homes can address (e.g., [11]), but there have been few widespread successes in specific interactions in that area.

FIRST EVALUATION OF PRE-PATTERNS

We evaluated the effectiveness of our design patterns by having designers use them in evaluating and designing location-enhanced applications. In this section, we describe our first round of evaluation.

Participants

Nine pairs of designers (18 designers total) participated in the first round. Four pairs were professionals, and the other five pairs were graduate students in the School of Information Management and Systems at UC Berkeley. These professional pairs had an average of 8½ person-years of experience combined (ranging from 6 to 10 combined), while the student design pairs each had an average of 4½ person-years of combined experience (with one pair having 8, the others having at most 3).

The pairs were divided into two categories based on experience. High-experience pairs had at least 6 person-years of combined experience, and low-experience pairs had at most 3. They were also divided into two conditions, 4 with patterns and 5 without (see Table 1).

We emailed our design patterns to the groups in the patterns condition two days beforehand so that they could familiarize themselves with the patterns. Designers in this

	High experience	Low experience
Patterns	4, 9	5, 6
No Patterns	1, 2, 7	3, 8

Table 1. Design pairs by condition for our first round of evaluation. For example, design pairs 4 and 9 had high levels of experience and were in the patterns condition.

	Patterns useful for Evaluation Task	Patterns useful for Design Task	Patterns useful for Other Projects
4-1	2	4	3
4-2	4	4	3
5-1	2	5	4
5-2	4	3	3
6-1	3	2	4
6-2	1	2	4
9-1	4	4	4
9-2	5	5	5
Avg	3.1	3.6	3.8
Stdev	1.4	1.2	0.7

Table 2. Feedback about our design patterns from participants in the patterns condition (1–5, 5=high)

Condition	Creativity	Completeness	Quality
<i>All pairs</i>			
Patterns	$\mu = 5.08$ $\delta = 1.24$	$\mu = 5.42$ $\delta = 1.00$	$\mu = 4.67$ $\delta = 1.07$
No patterns	$\mu = 4.00$ $\delta = 1.31$	$\mu = 4.67$ $\delta = 1.40$	$\mu = 4.53$ $\delta = 1.51$
<i>Pairs with low experience</i>			
Patterns	$\mu = 5.17$ $\delta = 0.75$	$\mu = 5.50$ $\delta = 1.22$	$\mu = 4.83$ $\delta = 1.17$
No patterns	$\mu = 3.83$ $\delta = 1.17$	$\mu = 4.00$ $\delta = 1.79$	$\mu = 4.00$ $\delta = 1.79$
<i>Pairs with high experience</i>			
Patterns	$\mu = 5.00$ $\delta = 1.67$	$\mu = 5.33$ $\delta = 0.82$	$\mu = 4.50$ $\delta = 1.05$
No patterns	$\mu = 4.11$ $\delta = 1.45$	$\mu = 5.11$ $\delta = 0.93$	$\mu = 4.89$ $\delta = 1.27$

Table 3. An analysis of the judges' ratings of the designs, on a scale of 1–7 (7=high). The judges on average rated the pairs who had patterns higher than those who did not, in creativity and completeness, and in quality except for pairs with high experience.

condition were also provided paper copies of the patterns during the session and given a few minutes at the start of the evaluation to look them over.

Method

The evaluation consisted of two tasks. The first task explored to what degree patterns assisted designers in evaluating an existing design. The designers performed a heuristic evaluation for 30 minutes on a design for a location-enhanced bus locator. The design consisted of textual descriptions of what a user can do with the service and storyboards that illustrated how the user could interact

with it. The design pairs were asked to go through these mockups, circling any problems they found and rating the severity of the problem. Later, we compared the heuristic evaluations between the two conditions to see if there were any significant differences in the types of errors found.

The second task was to design a location-enhanced service to help customers in a shopping mall. The designers were given a description of what services the mall would like, and could design other complementary services if desired. They were told they could make any assumptions they thought were reasonable, and could use any technologies they thought would be available within the next few years. They were given 80 minutes to create a design, using pens, paper, post-it notes, or a whiteboard. Afterwards, they had another 10 minutes to present their designs to us as if we were their client. We videotaped the design and presentation sessions and later reviewed the tapes to see how the design patterns affected the design process. Specifically, we looked for evidence of the following:

- Are patterns useful for introducing designers to ubicomp?
- Are patterns useful for communicating between designers? For example, do designers adopt the pattern language vocabulary as they talk about a design?
- Are patterns useful for creating designs?
- Are patterns useful for creating higher-quality designs?

Participant Feedback

After finishing both tasks, the designers filled out a questionnaire asking for basic demographic information, what type of design background they had, and whether they had designed a location-based service before. All of the design pairs had GUI and web design experience. Three of the nine pairs had designed location-based services before.

The design pairs in the patterns condition were also asked whether they had used design patterns before, and to rate the usefulness of our design patterns for the evaluation task (task 1), the design task (task 2), and other projects they might do in the future. This was done on a five-point scale (1=low and 5=high). The results are summarized in Table 2.

Overall, the 4 design pairs that used our patterns rated them 3.6 of 5 for usefulness in the design task. 5 out of the 8 participants gave a 4 or 5 rating. The other 3 gave a 2 or 3 rating. Two of these said there was not enough time to absorb the patterns, and one said that they were a little hard to understand because English was his second language.

There was no consensus on the usefulness of patterns for evaluation, with ratings fairly evenly distributed. We observed that the pairs only used patterns minimally during the evaluation. Finally, the participants thought overall the design patterns would be useful for future projects (3.75 out of 5, no one ranked below 3).

Judging

We also wanted to know if patterns are useful for creating higher-quality designs. To do this, we recruited three HCI

graduate students familiar with ubiquitous computing to judge the designs. For each design pair, the judges rated the design on creativity, completeness, and quality on a seven-point scale (1=low and 7=high). The judges watched each of the ten-minute presentations, without knowing which pairs were in which condition. The videos were shown in a different order to each judge to minimize bias. We then averaged the scores for each question across the judges.

Although the results are not statistically significant, possibly due to the low number of judges and low number of participants, the judges on average rated the pairs who had patterns higher than those who did not, in creativity and completeness. They also rated them higher in quality, except for pairs with high experience. See Table 3.

First Evaluation Observations

During the design tasks, we observed several themes.

Patterns Helped Novice Designers

Unsurprisingly, pairs with the least number of years of design experience struggled the most with understanding how to apply new technologies in solving problems. For example, design pairs 3 and 8 (both in the no-patterns condition and having little design experience) had difficulty with understanding the capabilities and limitations of devices, how a location-enhanced application might work, and what kinds of features such an application might offer.

However, design pairs 5 and 6, who were in the patterns condition, had a comparable number of years of design experience to design pairs 3 and 8, but did not face these same difficulties. Pair 5 had no experience in designing a location-based service, but extensively used the patterns in generating ideas and finding solutions. Pair 6 did have some experience in creating location-enhanced applications and had some knowledge of ubiquitous computing research, but still found the patterns useful in coming up with new ideas and in explaining ideas to one another.

Patterns Helped Designers with Unfamiliar Domain

We also observed that design pairs could quickly make use of our patterns for a domain that they were unfamiliar with. For example, neither of design pairs 4 and 5 had ever designed a location-enhanced application before, but both pairs made extensive use of the patterns to generate new ideas and to communicate with one another. It was common to see one person leafing through the patterns and skimming through the names and pictures to come up with ideas. It was also common to see one designer show the other a pattern to help explain a particular concept. Design pairs 6 and 9 only made modest use of the design patterns, but also did not encounter any difficulties using them.

Patterns Helped Designers Communicate Ideas

We expected designers to use the names of the patterns when they were communicating with one another, but it turns out that very few of these names were actually said out loud. More often, designers used the patterns to communicate ideas by pointing at a particular picture. We

believe that this is because location-enhanced applications are a new domain with few well-established terms. A pattern language could help foster the adoption of such terms, but in retrospect it was unrealistic to expect designers to adopt these terms in a short design session.

However, one interesting observation is that all of the design pairs used familiar web metaphors in describing their ideas, such as “pages”, “cookies”, and “bookmarks”, as well as the hierarchical organization found in Yahoo and shopping options found on Amazon. Designers in both conditions were implicitly using design patterns that they had direct experience in actually using or had previous experience in designing. This common grounding helped designers express ideas quickly and concisely.

Patterns Helped Designers Avoid Some Design Problems

We also observed that some design pairs in the non-patterns condition often struggled to find solutions, spending a lot of time on cases that we had patterns for. For example, design pair 3 spent a large amount of time coming up with what should be displayed on an Active Map (a map that displays the user’s current location and nearby points of interest) and how it would actually work. Active Map (B1) is one of the patterns in our pattern language, and was one that was used by all of the design pairs in the patterns condition.

Design pair 7, also in the no-patterns condition, faced a related problem. As professionals, their design was quite extensive and had many interesting ideas for optimizing shopping time and creating wish lists while at the mall. However, midway through, one of the designers started disliking the amount of control the application had, saying, “This is really cool and efficient, but I kind of just want to wander around.” This was an issue that we actually addressed in the pattern Serendipity in Exploration (D5).

These two examples point to a deeper issue about patterns. Many of the designers actually came up with the same ideas, such as using a map to show a person’s current location and having comparison shopping. However, one difference is that design pairs in the no-patterns condition had to revisit and sometimes fix design decisions more often than design pairs in the patterns condition. For example, design pairs 3 and 8, both in the non-patterns condition, both came up with a kiosk design that they changed midway through the design task, before changing to a PDA design that was better suited for the task.

We believe this is because our patterns represent solutions that others have thought through. Our patterns encapsulated knowledge about how a solution could be used on a particular device, how it might work, and how it could be presented to end-users. In contrast, designers in the no-patterns condition often had to come up with solutions from scratch, and sometimes overlooked important tradeoffs.

Patterns Did Not Help With Privacy

As expected, nearly all of the design pairs identified privacy as a design issue. In all of these cases, the design pairs

struggled with this issue for a while and then set the issue aside to work on the main functionality. In all cases, privacy was treated as a secondary issue. In many respects, this matches the evolution of web design. Very few early web sites addressed privacy in any meaningful way.

Unfortunately, none of the design pairs managed to use our privacy patterns in any meaningful way. We believe this happened for three reasons. First, we did not emphasize privacy sufficiently in the higher-level patterns. Second, the privacy design patterns are relatively abstract and do not lend themselves well to visual representations. As noted earlier, our participants typically leafed through the patterns to generate ideas, and hence visual representations of actual solutions worked best. Third, privacy is an abstract concept that can only be made concrete in the context of an actual task [17]. It does not make sense to talk about privacy itself, but rather how a specific design supports or inhibits privacy. Our patterns discuss privacy in an abstract manner, making it harder to find direct solutions to problems.

This last point also underscores a subtle issue here with respect to privacy, which is that customers are unlikely to judge an application based on its privacy merits alone. As noted by Whitten and Tygar, security is a secondary feature that people expect in the context of a task [25]. The same is true for privacy. Thus, it makes sense that designers would focus first on functionality and second on privacy.

Designers Generally Liked the Patterns

In general, designers who were in the design patterns condition did like the patterns. One designer said, “Good idea to identify design patterns for ubicomp.” However, one problem was that there were “too many patterns to digest”. This designer summarized his perspective on our patterns by saying, “If we had more time, I’m sure that we would be able to use these patterns to tailor them to our own ideas.”

SECOND EVALUATION OF PRE-PATTERNS

Based on the first round of evaluations, we edited their content to make them easier to learn and reduced the number of pre-patterns to 30 for the second round. We recruited seven pairs of designers for this round. Three pairs were professionals, and the other four pairs were graduate students from local universities. All but one pair had access to our pre-patterns (along with the 5 pairs in the non-pattern condition from the first round, this results in 6 pairs in each condition for this evaluation). All of the professionals had a high experience level, and all of the students were

novices. We also modified the methodology, removing the heuristic evaluation, adding 15 minutes before the design task to read the patterns, and adding a short 10-minute quiz to ensure that the designers were familiar with the patterns. This approach made it easier for the design pairs to familiarize themselves with the patterns before doing the design task.

Participant Feedback

9 out of 12 designers in the pattern condition felt that the design patterns helped with the design task, and 11 out of 12 felt that the design patterns would help with their future work. We also received stronger positive feedback regarding the patterns. One designer said, “These patterns are almost like a checklist. You can cover all of your bases.” Nearly all of the designers at the end of the study expressed interest in our patterns.

Judging

To judge the designs in the second round, we recruited a student who was a teaching assistant for an undergraduate HCI class, and two researchers familiar with ubicomp. For each pair, the judges rated the design on how much they agreed with ten statements, such as “Using this device would enable me to accomplish tasks more quickly,” and “I would find this device useful at the mall,” on a seven-point scale (1=low and 7=high). The judges watched each of the ten-minute presentations, without knowing which pairs were in which condition. We then averaged the scores for each question across the judges.

Although most of the results are not statistically significant,

Condition	Accomplish tasks more quickly	Privacy would not be compromised	Make product & store info more available	Useful
<i>All pairs</i>				
High experience	$\mu = 5.50$ $\delta = 0.55$	$\mu = 4.28$ $\delta = 1.18$	$\mu = 5.33$ $\delta = 0.67$	$\mu = 5.17$ $\delta = 0.51$
Low experience	$\mu = 4.22$ $\delta = 0.78$	$\mu = 4.44$ $\delta = 0.75$	$\mu = 4.44$ $\delta = 0.62$	$\mu = 4.06$ $\delta = 0.93$
<i>All pairs</i>				
Patterns	$\mu = 5.11$ $\delta = 1.07$	$\mu = 4.22$ $\delta = 0.96$	$\mu = 4.72$ $\delta = 0.88$	$\mu = 4.67$ $\delta = 1.26$
No patterns	$\mu = 4.61$ $\delta = 0.77$	$\mu = 4.50$ $\delta = 1.01$	$\mu = 5.06$ $\delta = 0.68$	$\mu = 4.56$ $\delta = 0.50$
<i>Pairs with low experience</i>				
Patterns	$\mu = 4.33$ $\delta = 0.88$	$\mu = 4.78$ $\delta = 0.77$	$\mu = 4.11$ $\delta = 0.69$	$\mu = 3.78$ $\delta = 1.26$
No patterns	$\mu = 4.11$ $\delta = 0.84$	$\mu = 4.11$ $\delta = 0.69$	$\mu = 4.78$ $\delta = 0.38$	$\mu = 4.33$ $\delta = 0.58$
<i>Pairs with high experience</i>				
Patterns	$\mu = 5.89$ $\delta = 0.51$	$\mu = 3.67$ $\delta = 0.88$	$\mu = 5.33$ $\delta = 0.58$	$\mu = 5.56$ $\delta = 0.19$
No patterns	$\mu = 5.11$ $\delta = 0.19$	$\mu = 4.89$ $\delta = 1.26$	$\mu = 5.33$ $\delta = 0.88$	$\mu = 4.78$ $\delta = 0.38$

Table 4. A subset of an analysis of the judges’ ratings of the designs, on a scale of 1–7 (7=high). Bold pairs of cells show significant differences (paired *t*-test, $p < 0.1$).

the judges overall rated novice pairs who had patterns lower than those who did not in 7 out of 10 questions. However, they rated expert pairs who had patterns equal or higher than those who did not in 9 out of 10 questions. They also rated expert pairs without patterns higher than novice pairs with patterns in all 10 questions. One possible interpretation is that having experience is more important than using patterns, but expert designers know how to apply patterns better than novices and therefore get more benefit from them. The few statistically significant results, which are shown in Table 4, also lend themselves to this interpretation.

Second Evaluation Observations

We had several interesting qualitative observations on the effects of the pre-patterns on design. More design pairs adopted the language of the patterns verbally than in the first round. Also, the design pairs often communicated their ideas through physical exchange of the patterns and by pointing to examples more readily than in the first round.

One pair mentioned that they used the pattern groups as “a way to organize their ideas.” Another pair drew inspiration from the Serendipity in Exploration (D5) pattern, stating that the location-based service they were designing “should not be a pushy salesperson but allow for free roaming.” A third pair used the patterns in an unanticipated way. Instead of simply culling ideas from the patterns, they annotated their designs with particular pattern references (e.g., writing “A1: Active Map” next to their sketched UI). One of the designers in the pair said, “It’s interesting because these [patterns] all sort of lay out the problem and the solution on a page, so just by saying that C2 is this one—it’s actually a quicker way of going through this whole procedure.”

However, the participants still failed to take advantage of the privacy patterns. 4 out of 6 pattern groups talked about privacy, but only one group actually used any of the privacy patterns directly, using three privacy patterns.

FUTURE WORK

In the future we will use feedback from the designers to make another iteration on our pattern language. We are especially interested in how to make the privacy patterns easier to understand and use. We will also continue our evaluations to further our understanding of how design patterns can help designers.

CONCLUSION

In this paper, we introduced the first pattern language for ubiquitous computing, consisting of 45 pre-patterns organized into four pattern groups. These pre-patterns discuss application genres, physical-virtual spaces, interaction and systems techniques for managing privacy, and techniques for fluid interactions. We also discuss what we believe is the first controlled study of design patterns with designers. We asked sixteen pairs of designers to design a location-enhanced application. We observed that patterns helped new and experienced designers unfamiliar with ubiquitous computing, in generating and communicating ideas, and in avoiding design problems early in the design process.

ACKNOWLEDGMENTS

We thank Quan Tran, Chris Beckmann, Jeff Heer, Alan Newberger, Ed de Guzman, Tara Matthews, and the rest of GUIR for their early feedback on our design patterns. This research has been funded by NSF (IIS-0205644).

REFERENCES

1. Third European Conference on Pattern Languages of Programming and Computing. <http://www.coldewey.com/europlop98/cfp.htm>
2. Alexander, C., *A Pattern Language: Towns, Buildings, Construction*: Oxford University Press, 1977.
3. Bayle, E., et al., Putting it all together: towards a pattern language for interaction design, *SIGCHI Bulletin*, vol. 30(1): pp. 17-23, 1998.
4. Borchers, J., *A Pattern Approach to Interaction Design*: John Wiley and Sons, 2001.
5. Casaday, G. Notes on a Pattern Language for Interactive Usability. In Proceedings of *Human Factors in Computing Systems: CHI 1997*. Atlanta, GA. pp. 289-290 1997.
6. Dearden, A., J. Finlay, L. Allgar, and B. McManus. Evaluating Pattern Languages in Participatory Design. In Proceedings of *Human Factors in Computing Systems: CHI 2002 (Extended Abstracts)*. Minneapolis, MN. pp. 664-665 2002.
7. Erickson, T., The Interaction Design Patterns Page. http://www.pliant.org/personal/Tom_Erickson/InteractionPatterns.html
8. Erickson, T. Lingua Francas for Design: Sacred Places and Pattern Languages. In Proceedings of *Symposium on Designing Interactive Systems 2000 (DIS2000)*. New York, NY. pp. 357-368 2000.
9. Gamma, E., R. Helm, R. Johnson, and J. Vlissides, *Design Patterns*: Addison-Wesley, 1995.
10. Graham, I., *A Pattern Language for Web Usability*. Reading, MA: Addison-Wesley, 2003.
11. Kidd, C.D., et al. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proceedings of *The Second International Workshop on Cooperative Buildings - CoBuild'99* 1999.
12. Landay, J.A. and G. Borriello, Design Patterns for Ubiquitous Computing, *Computer*, vol. 36(8): pp. 93-95, 2003.
13. Meszaros, G., A Pattern Language for Pattern Writing. 2000. <http://hillside.net/patterns/writing/patternwritingpaper.htm>
14. Nielsen, J., When Bad Design Elements Become the Standard. 1999. <http://www.useit.com/alertbox/991114.html>
15. Pfeiffer, E.W., WhereWare. *MIT Technology Review*, 2003: p. 46-52.
16. Rekimoto, J. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In Proceedings of *10th ACM Symposium on User Interface and Software Technology (UIST'97)*. Banff, Canada. pp. 31-39 1997.
17. Schwartz, B., The Social Psychology of Privacy. *American Journal of Sociology*, 1968. 73(6): p. 741-752.
18. Siegel, D., Set Your Link Colors Ergonomically. 1996. <http://www.dsiegel.com/tips/wonk3/links.html>
19. Tidwell, J., Common Ground: A Pattern Language for Human-Computer Interface Design. 1999. http://www.mit.edu/~jtidwell/common_ground.html
20. UsabilityNet, Card Sorting. <http://www.hostserver150.com/usabilit/tools/cardsorting.htm>
21. van Duyne, D.K., J.A. Landay, and J.I. Hong, *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Reading, MA: Addison-Wesley, 2002.
22. van Welie, M., Web and GUI Design Patterns. 2001. <http://www.welie.com/>
23. van Welie, M., K. Mullet, and P. McInerney. Patterns in Practice: A Workshop for UI Designers. In Proceedings of *Human Factors in Computing Systems: CHI 2002*. pp. 908-909 2002.
24. Weiser, M., The Computer for the 21st Century. *Scientific American*, 1991. 265(3): p. 94-104.
25. Whitten, A. and J.D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In Proceedings of *8th USENIX Security Symposium*. pp. 169-184, Aug 23-26 1999.

A – Ubiquitous Computing Genres	B – Physical-Virtual Spaces	C – Developing Successful Privacy	D – Designing Fluid Interactions
Describes broad classes of emerging applications, providing many examples and ideas	Associating physical objects and spaces with information and meaning; location-based services; helping users navigate such spaces	Policy, systems, and interaction issues in designing privacy-sensitive systems	How to design for interactions involving dozens or even hundreds of sensors and devices while making users feel like they are in control
Upfront Value Proposition (A1) Personal Ubiquitous Computing (A2) Ubiquitous Computing for Groups (A3) Ubiquitous Computing for Places (A4) Guides for Exploration and Navigation (A5) Enhanced Emergency Response (A6) Personal Memory Aids (A7) Smart Homes (A8) Enhanced Educational Experiences (A9) Augmented Reality Games (A10) Streamlining Business Operations (A11) Enabling Mobile Commerce (A12)	Active Map (B1) Topical Information (B2) Successful Experience Capture (B3) User-Created Content (B4) Find a Place (B5) Find a Friend (B6) Notifier (B7)	Fair Information Practices (C1) Respecting Social Organizations (C2) Building Trust and Credibility (C3) Reasonable Level of Control (C4) Appropriate Privacy Feedback (C5) Privacy-Sensitive Architectures (C6) Partial Identification (C7) Physical Privacy Zones (C8) Blurred Personal Data (C9) Limited Access to Personal Data (C10) Invisible Mode (C11) Limited Data Retention (C12) Notification on Access of Personal Data (C13) Privacy Mirrors (C14) Keeping Personal Data on Personal Devices (C15)	Scale of Interaction (D1) Sensemaking of Services and Devices (D2) Streamlining Repetitive Tasks (D3) Keeping Users in Control (D4) Serendipity in Exploration (D5) Context-Sensitive I/O (D6) Active Teaching (D7) Resolving Ambiguity (D8) Ambient Displays (D9) Follow-me Displays (D10) Pick and Drop (D11)

Figure 1. The overall table of our design pre-patterns for ubiquitous computing. These design patterns are organized into four pattern groups, sets of patterns that are related by a common theme. Generally speaking, lower-numbered patterns are higher-level, more abstract, and applicable in more cases than higher-numbered patterns. For example, Upfront Value Proposition (A1) is a high-level pattern that can be applied to a wide-range of applications, while Enabling Mobile Commerce (A12) is a pattern specific to that domain.

A12 • ENABLING MOBILE COMMERCE



Figure 1. Progress in sensors, small devices, and wireless networking is enabling new kinds of mobile commerce. This figure shows a small PDA with a barcode scanner. One could imagine using a system like this to get product reviews while out shopping, as well as recommendations and price comparisons.

• BACKGROUND

This pattern describes how businesses can use ubiquitous computing technologies to enable shoppers. It is an example of PERSONAL UBIQUITOUS COMPUTING (A2) and UBIQUITOUS COMPUTING FOR PLACES (A4) and should be used with an UP-FRONT VALUE PROPOSITION (A1).

• PROBLEM

How can businesses, both small and large, make use of improvements in sensors, devices, and wireless networking to make shopping more effective, more fun, and more enjoyable for their customers?

• SOLUTION

Businesses can provide mobile commerce services to their customers to make shopping easier, faster, and more fun. These kinds of

services are slowly emerging. Here, we describe some innovative ideas that are being rolled out.

One idea is to make it easy to find more information about products before you buy them. Figure 1 shows a small wirelessly-networked PDA that also has a barcode scanner. People can scan in barcodes to immediately find reviews, making them more confident about their purchases. Such a service could also provide recommendations (and depending on who offers the service and what their business model is, possibly price comparisons as well).

Figure 2 shows FastFrog, an idea tried out in several metropolitan areas in late 2000. Shoppers could use a small handheld to create wish lists that could be shared with friends and family.



Figure 2. FastFrog lets people build a wish list by scanning items in stores using a handheld. This wish list can be viewed on the FastFrog web site, and can be emailed to friends and family for birthdays, anniversaries, and other occasions.

As location-based services are slowly rolled out, mobile commerce GUIDES FOR EXPLORATION AND NAVIGATION (A5) will likely become prevalent. Figure 3 shows a small PDA application that presents an ACTIVE MAP (B1) of a shopping mall. This map could show where the user currently is, make it easy to FIND A FRIEND (B6), as well as interesting stores and sales.

Figure 2. Each pre-pattern has a number (e.g., A12), name (“Enabling Mobile Commerce”), sensitizing image, background that relates this pattern to other patterns, problem statement, solution, and references.

A1 • UP-FRONT VALUE PROPOSITION



Figure 1. TrafficGauge offers a PDA that provides up-to-date traffic information for the Seattle area. This is a clear and compelling value proposition for users.

• BACKGROUND

An application needs to communicate its purpose immediately and clearly. This pattern describes how to come up with that message, and can be applied to any application, including PERSONAL UBIQUITOUS COMPUTING (A2), UBIQUITOUS COMPUTING FOR GROUPS (A3), and UBIQUITOUS COMPUTING FOR PLACES (A4).

• PROBLEM

Some applications do not make it obvious what they offer users. This is critically important for ubiquitous computing applications, since there are many fears that emerging applications will be used to track users and invade their privacy.

• SOLUTION

Each application needs to offer a clear value proposition that states how it can help users. The value proposition can be a persuasive promise or a unique benefit that can greatly help the user in some way. Here are some hypothetical value propositions.

- Emerging cell phone services offer ACTIVE MAPS (B1) of your current location, promising that you will never be lost again.

- Motion sensors connected to the heating and ventilation systems in SMART HOMES (A8) offer energy efficiency for home owners.
- An electronic message board can be set up in SMART HOMES (A8), making it easy for family and friends to share photos and scribble messages to one another.
- Key chains will act as PERSONAL MEMORY AIDS (A7), recording their current location when you lock your car doors so you will always be able to find where you parked.
- Location-tracking systems for hospitals can make it easier for nurses and doctors to find each other and help ensure high-quality patient care.



Figure 2. Emerging cell-phone applications let people search for nearby businesses and restaurants, and provide maps of how to get there from one's current location. This is a simple but persuasive value proposition: "never get lost again."

These applications offer solutions to a wide range of needs, from safety to convenience, from efficiency to making it easier to stay in touch with family and friends. *The key here is that they can be quickly and easily explained.*

The last example, location-tracking for hospitals, is an especially important one, as it is in a workplace setting. There are many legitimate fears that such a location-tracking system could also be abused by management, for example by monitoring how often nurses go to the bathroom and by checking how long they spend with each patient.

Draft of Design Patterns for Ubiquitous Computing (Send comments to [\[redacted\]](mailto:))

B6 • FIND A FRIEND



Figure 1. AT&T Wireless' mMode service allows customers to add friends to a friend list, find out who is nearby, and call or send messages to them. Users can make themselves invisible whenever they want.

• BACKGROUND

This pattern discusses services that allow people to find where their friends are while allowing those friends some level of privacy. This pattern is useful for GUIDES FOR EXPLORATION AND NAVIGATION (A5).

• PROBLEM

People would like to know where their friends are, for impromptu communication and gatherings. At the same time, those people may not always want to be tracked.

• SOLUTION

Finding friends can be useful in many situations, such as coordinating at conferences, finding lost friends in large crowds, or making sure someone got home safely.

There are two primary design issues when designing find-a-friend services: how to display friends' locations, and how to manage people's privacy concerns.

Displaying people's location • There are several different ways of displaying a person's location. A straightforward approach is to simply show the location in text, for example "near corner of Euclid Ave and Hearst Ave" or "in Soda Hall". Another approach is to show the data on a map, or possibly even an ACTIVE MAP (B1) that is constantly updated.



Figure 2. UC San Diego's ActiveCampus project shows your friends' location in real time. While useful, this visualization raises many privacy concerns.

Managing privacy concerns • There are many privacy concerns about find-a-friend applications due to the potential for abuse. This is not just the fear of "Big Brother," but also so-called "Little Brothers," including phone companies, friends, parents, and jealous boyfriends and girlfriends.

One way of managing these concerns is by giving users the ability to LIMIT ACCESS TO PERSONAL DATA (C10). The most common approach here is to have a buddy list that lists which friends can see your current location.

C9 • BLURRED PERSONAL DATA



Figure 1. This mockup of the Place Bar browser component would let users select the level of location information disclosed to web sites, potentially on a page by page basis, as part of their browsing activity.

• BACKGROUND

One aspect of making people feel that their privacy is being preserved is by providing a REASONABLE LEVEL OF CONTROL (C4) over one's personal information. This pattern looks at how blurring personal data can give users control.

• PROBLEM

How can people share personal information such as location and activity without revealing all of the details?

• SOLUTION

Here we describe two different ways for blurring data, removing details while preserving some level of utility.

Blur individual pieces of data • One way of sharing dynamic personal information such as location and activity is to provide a *blurred* version of it, which is still accurate but at a coarser granularity. Figure 1 shows a mockup of the Place Bar, a component for web browsers that would let end-users share their current location with web services at a level they are comfortable with. The finest level of detail is Soda Hall room 525, while the coarsest level of detail is the city of Berkeley, California.

Figure 2 shows how blurring can be applied to streaming video to preserve privacy. There is enough information to see that there are people, but not necessarily who those people are.



Figure 2. Blurring can also be applied to video to de-identify individuals and to mask activities.

Blur by aggregating multiple pieces of data • Data can also be blurred by aggregating it with similar data. For example, location data for a single individual can be aggregated spatially ("You have been to this restaurant five times this past year, most recently on November 7") or temporally ("You were around San Francisco on November 7") rather than keeping exact location data for each hour.

Data can also be aggregated by combining the data of several individuals, making it harder to identify the activities of any single person. For example, this is often done with medical records, providing analysts statistical data about patients without revealing precise data about any individual.

• REFERENCES

- Boyle, M., Edwards, C., and Greenberg, S. The Effects of Filtered Video on Awareness and Privacy. In the Proceedings of CSCW 2000. <http://www.cpsc.ucalgary.ca/roughlab/papers/2000090-Filtered-Video-CSCW00-Filtered-Video-CSCW00.pdf>. This paper discusses how effective different filters were in providing awareness while preserving privacy.
- Hong, J. I., Borelli, G., Landay, J.A., McDonald, D., Schilit, B., and Tygar, D. Privacy and Security in the Location-enhanced World Wide Web. In Ubicomp 2003 (Workshop on Ubicomp Communities: Privacy as Boundary Negotiation). <http://www.cs.berkeley.edu/~jasonh/papers/ubicomp2003-privacy-ubislab.pdf>. This workshop paper looks at a way of providing global location data in a simple, inexpensive, and privacy-sensitive manner. It also describes the Place Bar, shown in Figure 1.
- Sweeney, L. Guaranteeing anonymity when sharing medical data, the datally system. Proceedings, *Journal of the American Medical Informatics Association*. Washington, DC: Hanley & Belfus, Inc. 1997. <http://lab.uci.edu/education/sweeney/records/sweeneyatitah.html>. This system looks at how to share aggregated medical data.

D5 • SERENDIPITY IN EXPLORATION



Figure 1. Design for serendipity when users want to be spontaneous.

• BACKGROUND

An example of KEEPING USERS IN CONTROL (D4), serendipity in exploration is often used in touring and guidance applications, where services like USER-CREATED CONTENT (B4), and FIND A FRIEND (B6) can enhance such services.

• PROBLEM

Some people enjoy directed tours, but others like to just explore freely. Applications that are constrained to pre-determined courses may make people feel that they are not in control and end up being frustrated.

• SOLUTION

Design your applications to be flexible and controllable. Allow users to be spontaneous in their decisions. They may sometimes wish to end up in a specific location but want to be flexible in the path they take to get there.

For example, rather than providing an omniscient overhead map, a conceptual design

called the (De)Tour Guide lets people discover districts and landmarks "only upon approach, as if by chance." It also lets people "get lost on purpose, or to follow the idiosyncratic paths of unusual strangers."

Another way of designing for serendipity is to empower users with USER-CREATED CONTENT (B4). Rather than being just consumers of content, they can also be creators of TOPICAL INFORMATION (B2).

• REFERENCES

- Kasman, E. (2003). "User needs for location-aware mobile services." *Personal Ubiquitous Computing* 7: 70-79.

Figure 3. Some more examples of our pre-patterns for ubiquitous computing.