Rapid Prototyping and User-Centered Design of Interactive Display-Based Systems

Without investigation of real use, technical feasibility can be meaningless. Adopting multiphased, user-centered prototyping approaches—involving deployment over time outside the lab—can produce better functional systems.

Dan Fitton, Keith Cheverst, Chris Kray, Alan Dix, Mark Rouncefield, and George Saslis-Lagoudakis
Lancaster University

Within the manufacturing and engineering industries, *rapid prototyping* is the relatively fast physical fabrication of a design or concept for purposes such as demonstration, evaluation, or testing. Software engineers also use rapid prototyping to uncover requirements by analyzing prototypes provided early in the development process and gathering feedback. When developing ubiquitous computing systems, which often include hardware, software, and (of course) human factors, these two uses of rapid prototyping form an ideal design and development methodology.[1]

Much work on display-based ubicomp systems (and rapid prototyping in general) focus on producing proof-of-concept demonstrators, usually to gauge technical feasibility and collect initial user feedback.[2] In our work, we've found that it's often equally important to investigate factors such as use and appropriation[3] and that in some cases, without user studies, technical feasibility can be meaningless.

We used rapid prototyping combined with a phased, iterative, and user-centered design approach to develop five display-based ubicomp systems for real-world use over time. In this article, we discuss our aims, approach, and lessons learned.

Prototyped systems

The ubicomp systems that we developed and deployed use rapid prototyping techniques and a user-centered design approach.

Hermes 1

Hermes[4] is a system of interactive office door displays (shown in figure 1a), which provides asynchronous messaging facilities. We intend these displays to augment rather than replace existing messaging practices, such as the use of Post-it notes.

Aims. In developing the Hermes 1 displays, we wanted to explore issues of adoption and *situated interaction* in a location with both public and private elements (that is, outside an office door). We aimed to evaluate the system in place, allowing it to evolve as further requirements emerged over a significant period of time (months and years rather than the usual days or weeks) with daily use from a group of real-world users.

Approach. For Hermes 1, we used prototyping and early deployment, including only a small set of features from those technically possible. The minimized functionality let us concentrate on ease

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/05/$20.00 © 2005 IEEE

of use and reliability, and the relatively low complexity meant that we could quickly develop and refine prototypes (over a few days or weeks). From an early stage, we logged usage of our deployed prototypes.

We employed a *phased-development approach*,[4] where each phase (of approximately four months) had a different primary objective and involved several prototyping iterations. The first phase aimed to develop core functionality and involved deploying door displays outside the offices of two of the system's developers. This ensured some initial testing outside the lab but with users who wouldn't let low reliability and initial problems affect future use. From this initial phase, we uncovered a major issue impacting reliability: humans block the wireless network signal when standing in front of a Hermes 1 door display.

Phase 2 aimed to increase system reliability, solving problems encountered during phase one and deploying three additional door displays.

Phase 3 added new interaction methods (in response to user feedback) and increased the number of deployed door displays from five to 10. An interesting trade-off was that users were prepared to forgo security almost completely to reduce interaction time.

To avoid including only techies in our experiment, we gave door displays to two departmental secretaries (in phases 2 and 4) and a sociologist (in phase 4).

Later phases added new features and further improvements to reliability. Prototype deployment, evolution, and use lasted nearly three years, only ending when we moved to a new computing building.

Summary of development strategy. We conducted our initial feasibility study and requirements gathering by testing early prototypes with friendly users. We later expanded deployment to a broader range of users. We split the development process into phases, allowing for changes in the development focus. We split the phases into prototyping cycles, each lasting a few days or weeks. Our user-centered design combined with rapid prototyping encouraged adoption and appropriation by maximizing usability. We collected feedback from users both formally (questionnaires and interviews) and informally (email, complaints, and chance conversations) and logged all user interactions.

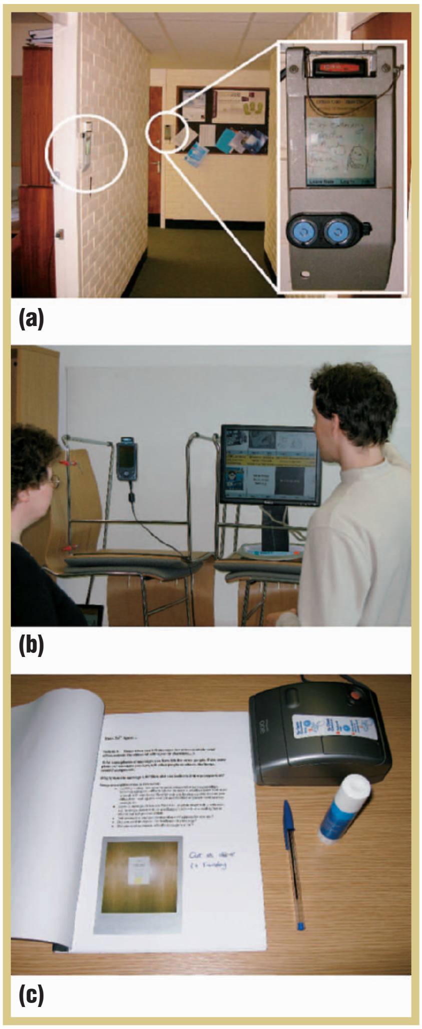


Figure 1. (a) Hermes 1 deployment, (b) a Hermes 2 showroom configuration, and (c) a Hermes 2 probe pack.

SPAM

The SPAM (SMS Public Asynchronous Messenger) system provides a lightweight alternative to the usual workplace communication methods (telephone, fax, and email). We developed the system to help support coordination across two sites in northern England that provide housing for former psychiatric patients. SPAM enables asynchronous short-message-service messaging between remote SPAM displays and between SPAM displays and the mobiles phones of staff and patients.

Aims. First, we wanted to develop and deploy a reliable, easy-to-use communication system based on SMS to enable messaging to a place rather than a person. This was motivated by engaged phone lines causing communication problems. Also, we wanted to help support coordination with staff members working with patients off site.[5] Second, we wanted this system to help us understand this domain further; logging user interaction for later analysis was crucial.

Approach. Our initial approach involved a Participatory Design workshop,[6] where we used chosen scenarios (informed by previous ethnographic work) and props (including a Hermes display) to generate discussions about requirements and possible technology solutions.

Following the workshop, we rapidly prototyped a system based on these requirements using mainly off-the-shelf hardware and software. This let us create a prototype solution quickly (in approximately one month). We then spent approximately a month on testing and burn-in to ensure the system had strong reliability—an absolute necessity given the deployment domain. Immediately following testing, we deployed the prototype systems at two locations. We quickly received feedback from users, which we used to drive minor modifications—for example, blocking senders (patients) who sent offensive messages.

Figure 2. Hermes photo display deployment and user interface.

Summary of development strategy. We gathered initial requirements at a design workshop, using rapid prototyping of scenarios (rather than hardware and software) with potential users. We then created rapid prototypes using mainly off-the-shelf hardware and software components to quickly produce a system ready for deployment. We included logging facilities in the prototype to enable use analysis.

Hermes 2

Following the success of Hermes 1 and SPAM,[5] we're currently aiming to understand how networked displays' physical placement and design in semi-wild settings influence and facilitate collaboration and community. To do this, we plan to redesign the original Hermes system and improve its underlying technology. We hope to deploy 40 door displays in Lancaster University's new computing department.

Aims. In Hermes 1, we used a one-size-fits-all approach for door display configuration (see figure 1a); with the forthcoming redeployment, we can investigate what users find acceptable in terms of parameters such as display size, number, housing, supporting infrastructure, and UI layout. At this early stage, we have two initial aims: to determine the physical form factor and the display configurations users desire.

Approach. To investigate our first aim, we've designed six hardware and UI configurations for door displays, including a range of sizes and display types. We rapidly prototyped these configurations using the appropriate off-the-shelf hardware and software. We mounted and arranged them to give an impression of how they might appear when deployed (see figure 1b), simultaneously providing showcase scenarios for all six potential door display configurations. We then gave 10 future door display owners a semistructured tour around the showcases, which we videotaped. We explained each configuration in the context of a scenario to highlight potential use. The experiment seemed to engage the participants, and having different prototypes next to each other enabled users to pick out the configurations they preferred—a typical comment being "I like that [user interface], but I'd want it on that display."

To investigate our second initial aim, we've given *probe packs*[7] to all participants in the showcase experiment. These packs contain a diary, instant camera, pen, and glue (see figure 1c). The diary is for recording messaging activities over a period of seven days, with pages for each day to record and describe messages they'd left for others and for themselves as well as messages left by others for them. The camera lets participants take pictures of messages, which they can glue into the diary and annotated appropriately to describe its context. Additionally, we did some paper prototyping of UI configurations with limited success but not as extensively as systems elsewhere have done.[8]

Summary of development strategy. We created and displayed rapid prototyping of multiple potential hardware and UI configurations. We gave potential users a guided tour, letting them choose which configuration they would prefer.

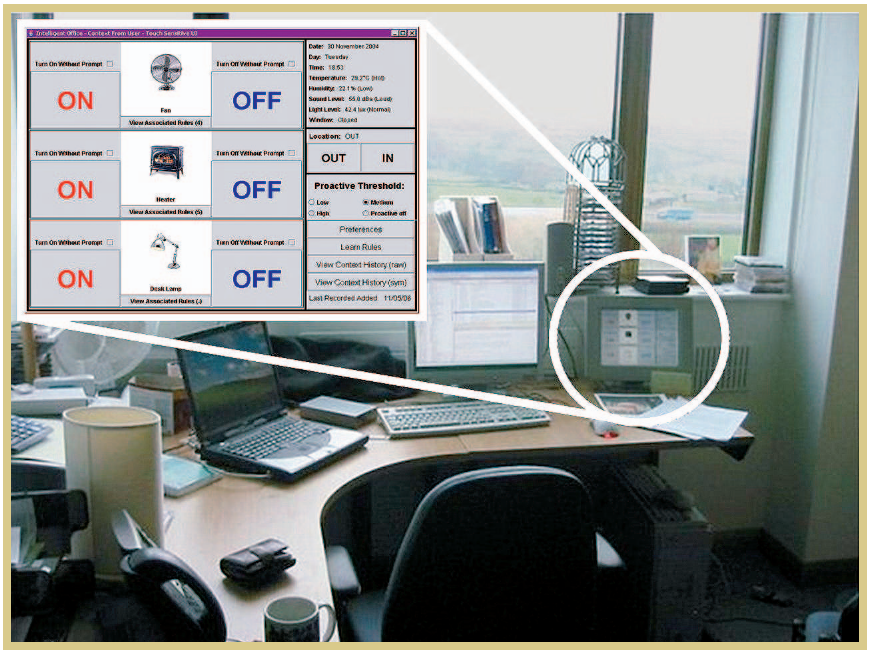


Figure 3. Intelligent office deployment and user interface (insert).

We gave probe packs to participants in the showcase experiment to uncover requirements and provide materials for a forthcoming design workshop.

Hermes Photo Display

The Hermes Photo Display[9] extends functionality provided in Hermes 1, enabling context sharing using pictures. It lets users send pictures both locally and remotely (using methods such as multimedia messaging, email, and Bluetooth) to a location where they're organized into a presentation and appear on an appropriate display (see figure 2). We hope to support messaging to a place rather than a person to promote a sense of community.

Aims. We aim to explore issues of use and user acceptance with this type of shared-picture-messaging display. To achieve this, we're investigating areas such as interaction methods and information presentation.

Approach. To date, our approach has involved a single display's rapid prototyping and deployment, focusing on methods of asynchronous and synchronous interaction through mobile phones. We initially deployed the display for use by a group of "friendly" users. Once reliability reached an acceptable level, we redeployed the system for real-world users (see figure 2). To elicit feedback from users in this public setting, we advertised for volunteers who we then asked to carry out prescribed tasks with the display and fill in a questionnaire.

Summary of development strategy. We investigated the feasibility of a public picture-messaging display in terms of the technology supporting the presentation of information and interaction methods. We used off-the-shelf and custom hardware and software to assess feasibility and enable deployment in our rapid prototyping. We investigated usage through logging and a questionnaire-based user study.

Intelligent Office user interface

The Intelligent Office system[10] controls various electrical appliances (such as a fan, heater, and lamp) by sensing environmental context and making proactive suggestions. The system makes these suggestions on the basis of rules it has automatically learned from a context history or from user input. For example, if it's hot and the fan is off, it would prompt you to turn on the fan. However, such a system's adoption and use raises many previously unaddressed HCI challenges.

Aims. The work aimed to investigate how to make the reasoning behind the Intelligent Office system's proactive behavior more visible to users, helping them to better understand it. This required a major UI redesign to provide a peripheral interactive desktop display (see figure 3), enabling manual control of appliances and information presentation as well as providing a platform for exploring the system's original goal.

Approach. We initially used rapid prototyping and small-scale deployment to gather requirements. Feedback from the previous UI helped guide this process. Development primarily involved only the UI, leaving the Intelligent Office system unchanged. We used a questionnaire to help generate system requirements, presenting questions in the context of possible scenarios.

Summary of development strategy. We used rapid prototyping of a UI for an intelligent office control system to explore improving the visibility of the system's proactive decisions for the user. High-fidelity prototyping helped us demonstrate the concept to potential users and, together with a questionnaire, enabled us to gather feedback and uncover requirements.

Issues

Building a novel ubicomp system typically involves dealing with a broad range of issues, often due to tailored hardware and specialized software solutions. We encountered several key issues in our use of rapid prototyping.

Tailoring off-the-shelf technology

Many times during our work we sought to tailor ordinary, off-the-shelf technology—for example, GSM modems, PDA devices, PC hardware, and a games console—to build our novel ubicomp prototypes. On the surface, this appears a practical solution, making use of readily available technology often with proven reliability at modest development cost. However, using this type of technology in ways the original designers might not have intended can have severe implications. In several cases, we found areas such as operating system and programming-language support missing or incomplete. Tailoring often requires extensive investigation and testing, greatly reducing the rapidity with which developers can produce prototypes.

When developing the Hermes Photo Display, enabling Bluetooth interaction using mobile phones proved far more time consuming than expected. After spending several weeks considering and testing various options, to gain the appropriate level of support and flexibility required, we had to use specific hardware, operating system, and programming APIs.

When attempting to support authentication using iButton readers on the Hermes 1 door displays, we discovered a complete lack of suitable solutions to support the required Java communications API on the platform we used. After spending weeks looking for an appropriate solution, a commercial solution became available. Unfortunately, after deploying prototype door displays with iButton support, we found this adversely affected the door displays' reliability, so we abandoned the idea.

While tailoring existing off-the-shelf technology seems appealing and can prove useful in some cases, we see a trade-off between the pitfalls of attempting to tailor existing technology and investing time and effort in a specialized solution. You must consider this trade-off carefully, especially in circumstances where you have limited or unavailable domain knowledge.

Tailoring often requires extensive investigation and testing, greatly reducing the rapidity with which developers can produce prototypes.

Rapid prototyping and reliability

The five systems we've described have high reliability requirements (but not necessarily at all phases of development), and this contrasts with the usual rapidity of short-term demonstration or testing. Reliability issues have often raised various kinds of challenges.

In the Hermes 1 system, where door displays were running 24/7, we found that display devices had unreliable operating system drivers and software components, occasionally causing a door display to crash after several weeks. This has implications for

- *Housing*. In the project's early stages, we had to forgo some security features of the housing to let us remove and reset the door display device (a PDA) easily.
- *Hardware platform*. To increase security, we had to modify door display devices to route the reset switch outside the case.
- *Operating system and software platform*. We had to temporarily decommission door displays to apply updates and change configuration settings to improve reliability.
- *Application*. We added sending a regular "heartbeat" from the client to the server to help us monitor whether a device was "alive" or not. The system used this information to notify a system administrator and the affected user via email.
- *User interface*. Occasionally a door display would crash but still display a UI, making it appear functional, confusing users, and reducing trust.
- *Human factors*. If someone wishing to leave a message on a door display doubts whether their message will actually reach its owner, he or she will simply use a Post-it note instead.

During the SPAM system's burn-in testing, we discovered that the GSM modems appeared to have a fault, occasionally causing them to stop responding until reset. Unable to replace these components at this late stage, we configured the display systems so that a single press of the power button would shut down the display application and power off. Then, when powered on, the system would automatically start the application and restore the previous state. SPAM system users understood this feature because other familiar information appliances, such as a mobile phone, might occasionally crash and similarly need to be turned off and back on.

We based the Hermes picture display on a Philips DesXcape Smart Display (a wireless Microsoft Windows XP Remote Desktop client), which enabled us to rapidly prototype and deploy a display system, requiring only power and wireless network coverage. However, on several occasions, disruption to the wireless link caused the display to unexpectedly disconnect from the Remote Desktop host. Also, the Remote Desktop connection caused additional overhead that seemed to make the host machine less stable. Using a Smart Display in this case demonstrated a trade-off between the need for reliability and for easy and fast deployment: requiring user feedback on the experience of using a high-fidelity prototype outweighed the minor reliability problems.

We concluded that rather than attempting to provide very high levels of reliability, it was more practical to provide failure notification and to manage user expectations.

Managing user expectations

In low-fidelity, inside-the-lab prototyping, encouraging users to suspend disbelief (or act as if a prototype is real and fully functional) is an accepted practice. However, most users expect prototype systems that are deployed externally and intended for everyday use to behave like finished products. This might be a reasonable expectation given that we're providing a system to be integrated within existing working patterns, effectively implying high reliability even though the system is only an experimental prototype.

Consequently, a crucial aspect of deploying in the real world is managing user expectations. We consider the development and maintenance of trust as a gradual process[11,12] and therefore added functionality slowly, in phases, to maintain strong reliability. We also tried to ensure that deployment stayed in line with system reliability, to avoid damaging the important initial trust-establishment process (by ensuring that users' first encounters with the prototype were successful). We haven't always succeeded with this. For example, during Hermes 1's later deployment phases, reliability problems affected user trust, and for some users, we found reestablishing trust to be challenging—despite strong improvements in the system's overall reliability.

Fostering a user-centered approach

During our work, it helped to use more traditional ethnographical methodologies to jump-start our user-centered rapid prototyping, gathering user requirements before the development process began (as we're doing with Hermes 2) to improve the initial prototypes' usability and acceptance. Effectively, this means that we'll need fewer rapid prototyping cycles at the beginning of the development process. Using a Participatory Design workshop in the SPAM system let us develop a successful initial prototype, which nearly met all its user requirements.

Time and again, the need has arisen for the designer to be sympathetic to users' patterns of use. For example, during Hermes 1's early phases of development, we forced owners to interact with their door display through a Web browser. This caused a problem for two reasons. It was often inconvenient for users to open a Web page and enter their user name and PIN to set or read a message. Additionally, and perhaps more importantly, the approach didn't fit with the way many users seem to process the task of leaving a message. Only by involving users in the system's design did we develop approaches that fit with their existing work patterns—for example, offering use of MSN Messenger and integrating Hermes 1 with their email systems.

Deployment for understanding domains

Recently employed in the Interliving Project,[13] *technology probes* are adaptations of *cultural probes* (which provide ethnographic data on usage) that help to inspire designers during the design process. These probes situate existing technologies in real homes, exposing inhabitants to new experiences, resulting in feedback different from what you might collect in a lab. While our use of prototype showcases does help provide user feedback to inspire the design process, we also use technology probes differently and perhaps more simply—we embed a logging system into the technology itself. Both Hermes 1 and SPAM included such logging of user interactions. Analyzing these logs improved our understanding of these two domains, helped drive the prototyping processes, and enabled us to quantify various aspects of usage.

We concluded that rather than attempting to provide very high levels of reliability, it was more practical to provide failure notification and to manage user expectations.

Lessons learned

Table 1 summarizes the different prototyping techniques we used in our user-centered rapid prototyping, along with their associated systems and key advantages and disadvantages. Additionally, it illustrates that we applied several techniques within each individual project, which has led to two broader central lessons.

Multiphased prototyping

In most cases, the faster or earlier you deploy a prototype, the faster you can obtain feedback and new requirements and improve the design. However, early prototypes are inherently unfinished and might have errors that, if you don't manage expectations, could negatively impact user attitudes toward the system and make user-centered design difficult.

Consequently, we use a multiphased prototyping approach, driven by user-

TABLE 1
Summary of techniques used.

Technique	Systems	Key advantages	Key disadvantages
Prototype showcases (demonstrating collections of potential prototypes to end users)	• Hermes 2	• Supports early user choice • Generates useful feedback • Users can signal the best parts of the different prototypes presented	• Time consuming to organize, design, and build potential prototypes • Difficult to preempt all concerns users might have
Participatory Design workshops (supporting the user-centered design of both early and more mature prototype systems)	• SPAM • Will also be used in Hermes 2 and specific Hermes Photo Display deployments	• Helps users feel truly involved in the user design process • Can use props and prototypes to inspire discussion	• Cost associated with organization of the workshop
Paper prototyping (demonstrating early user interface designs on paper to end users)	• Hermes 2	• Low cost • High flexibility • Helps include users early in the design process	• Can be difficult to provide sufficient fidelity for a user to appreciate the scenario
Questionnaire-based user studies (obtaining feedback on early or more mature prototypes)	• All	• Low setup cost • High flexibility	• Time consuming for user to complete a questionnaire properly • Can be difficult to investigate the rationale for a participant's response
Lab-based testing of early prototypes	• All	• Efficient and useful for ascertaining initial technical feasibility • Useful for burn-in reliability testing	• Open to requirements capture problem
Early deployment of prototypes			
Real-world users (unbiased)	• SPAM • Hermes 1 (second phase onward, where deployments involved end users not directly associated with the project) • Hermes Photo Display (second phase of deployment)	• Users can regularly experience real-world use of the prototype, generating useful feedback	• Difficult to maintain reliability, trust, and regular use • Usually requires some form of management and user support
Friendly users (connected with the work)	• Hermes 1 (initial phase where units were only deployed outside the developers' offices) • Hermes Photo Display (initial phase of deployment) • Intelligent Office • Will also be used in Hermes 2	• Users can experience real-world use with a potentially unreliable prototype without a large impact on trust (and use) • Generates useful feedback quickly	• Sometimes difficult to involve a range of users, so potentially open to the requirements capture problem • Sympathy is likely to affect judgment, so feedback may be biased, inaccurate, or both
Cultural probes (obtaining ethnographic data on usage)	• SPAM • Hermes 2 • Will also be used for specific deployments of the Hermes Photo Display—for example, to support specific societies and communities at Lancaster University	• Helpful insights into use • Relatively easy and fast to set up • Low financial cost	• Might not capture all use due to practicality of logging for participants
Deployed prototypes as technology probes (integrating logging facilities to help understand actual use and therefore help drive further developments)	• SPAM • Hermes 1 • Will also be used for Hermes 2 and certain deployments of the Hermes Photo Display	• Easy to collect data • Relatively easy to analyze and quantify use	• Implications for design • Analysis might include the time-consuming, manual data tagging

centric considerations. We apply different prototyping techniques at different project stages to gather user feedback about system properties, contextual factors, and their concerns. We then iteratively validate and refine a prototype until it meets the users' expectations and occasionally introduce new features to explore further options. In the case of prototype deployment, we distinguish phases in terms of the user groups exposed to the prototype. We incrementally move from fault-resilient users to sympathetic users before deploying it to real-world users. Combined with a longitudinal perspective (that is, evaluating prototypes for several months instead of just a few days), this approach lets us to gather more information than would have been possible otherwise.

Examples from our five projects that might have gone unnoticed using a conventional prototyping approach include the impact of the iButton hardware unreliability on user acceptance, usage issues with the initial UI in the Intelligent Office project, and the requirement for a blocking feature in SPAM.

Technical feasibility and use

In our work, exposing users to prototypes to generate feedback has been crucial to enable use. For example, the original Intelligent Office prototype demonstrated that such a system was technically possible (sensing environmental context and making proactive suggestions). However, after exposing the system to users, we found it was simply too difficult to use on a regular basis—the interaction model simply didn't fit the desktop environment in which it was used.

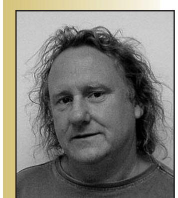
Another trade-off is in the use of off-the-shelf technology, which can provide benefits such as reducing development time and jumpstarting reliability, by using reliable building blocks. However, this type of technology typically must be tailored to a specific purpose, which we found sometimes difficult and challenging. When developing the Hermes Photo Display, we found that the Bluetooth discovery process is relatively long and often unreliable; during a user study with over 10 devices in range, the server had trouble discovering more than five or six at once.

the AUTHORS

Daniel Fitton is a research associate in the Department of Computing at Lancaster University. His research interests involve the design, rapid prototyping, investigation, and deployment of new and novel-situated display-based systems. He's also writing his PhD thesis, which focuses on the support of messaging and awareness with interactive situated technology. He received his BSc in computer science from Lancaster University. Contact him at the Dept. of Computing, InfoLab21, South Dr., Lancaster Univ., Lancaster LA1 4WA, UK; df@comp.lancs.ac.uk; www.dan-f.co.uk

Keith Cheverst is a senior lecturer in the Department of Computing at Lancaster University. His research interests involve context-aware and ubiquitous computing—in particular, computer-supported cooperative work and human factors issues. He received his PhD in computer science from Lancaster University. Contact him at the Dept. of Computing, InfoLab21, South Dr., Lancaster Univ., Lancaster LA1 4WA, UK; kc@comp.lancs.ac.uk.

Chris Kray is a research associate in the Embedded Interactive Systems group at the Department of Computing at Lancaster University. His research interests include ubiquitous computing, spatial reasoning, applied AI, and human-computer interaction. He's also working on a relative-positioning system and applications for pervasive display infrastructures. He received his PhD from Saarland University. Contact him at the Dept. of Computing, InfoLab21, South Dr., Lancaster Univ., Lancaster LA1 4WA, UK; kray@comp.lancs.ac.uk.



Mark Rouncefield is a senior research fellow in the Department of Computing at Lancaster University. His research interests include computer-supported cooperative work and the empirical study of work, organization, human factors, and interactive computer systems design. This strongly interdisciplinary work has led to extensive and continuing collaborations with colleagues in sociology, computing, informatics, and management. Contact him at the Dept. of Computing, InfoLab21, South Dr., Lancaster Univ., Lancaster LA1 4WA, UK; m.rouncefield@lancaster.ac.uk.

George Saslis-Lagoudakis is a PhD student in the Department of Computing at Lancaster University. He research interests include display-based ubiquitous computing systems and the interaction of personal mobile devices with such systems. He received his MSc in computer science from Imperial College, London. Contact him at the Dept. of Computing, InfoLab21, South Dr., Lancaster Univ., Lancaster LA1 4WA, UK; g.saslis-lagoudakis@comp.lancs.ac.uk.

We hope the experiences we've discussed will help other researchers and practitioners select an appropriate prototyping technique for future projects.

ACKNOWLEDGMENTS

The UK's Engineering and Physical Sciences Research Council's Caside (Cooperative Applications in Situated Display Environments) project (grant EP/C005589) funded this work, which builds on work carried out under the EPSRC-funded Equator and Casco (Context-Aware Support for Cooperative Applications in Ubiquitous Computing Environments) projects. The Science Research Investment Fund (SRIF II) e-campus initiative will provide hardware support for future trials based on the work described.

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, 1991, pp. 66–75.
2. J.F. McCarthy, T.J. Costa, and E.S. Liongosari, "UniCast, OutCast & GroupCast: Three Steps Toward Ubiquitous, Peripheral Displays," *Ubicomp 2001: Ubiquitous Computing*, LNCS 2201, Springer, 2001, pp. 332–345.
3. M.L. Markus and T. Connolly, "Why CSCW Applications Fail: Problems in the Adoption of Interdependent Work Tools," *Proc. Conf. Computer Supported Cooperative Work* (CSCW 90), ACM Press, 1990, pp. 371–380.
4. K. Cheverst, K.D. Fitton, and A. Dix, "Exploring the Evolution of Office Door Displays," *Public and Situated Displays:*

Social and Interactional Aspects of Shared Display Technologies, Kluwer, 2003, pp. 141–169.

5. K. Cheverst et al., "SPAM on the Menu: The Practical Use of Remote Messaging in Community Care," *Proc. Conf. Universal Usability* (CUU 03), ACM Press, 2003, pp. 23–29.

6. J. Greenbaum and M. Kyng, eds., *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, 1991.

7. W. Gaver, A. Dunne, and E. Pacenti, "Design: Cultural Probes," *Interactions: New Visions of Human-Computer Interaction*, vol. 6, no. 1, 1999, pp. 21–29.

8. J. Mankoff and B. Schilit, "Supporting Knowledge Workers beyond the Desktop with PALPlates," *Proc. Conf. Human Factors in Computing Systems* (CHI 97), ACM Press, 1997, pp. 550–551.

9. K. Cheverst et al., "Exploring Bluetooth-Based Mobile Phone Interaction with the Hermes Photo Display," *Proc. 7th Int'l Conf. Human Computer Interaction with Mobile Devices and Services* (Mobile HCI), ACM Press, 2005, pp. 47–54.

10. K. Cheverst et al., "Exploring Issues of User Model Transparency and Proactive Behaviour in an Office Environment Control System," to be published in *User Modeling and User-Adapted Interaction*, Kluwer, 2005.

11. K. Siau and Z. Shen, "Building Customer Trust in Mobile Commerce," *Comm. ACM*, vol. 46, no. 4, 2003, pp. 91–94.

12. E. Sillence et al., "Trust and Mistrust of Online Health Sites," *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI 04), ACM Press, 2004, pp. 663–670.

13. H. Hutchinson et al., "Technology Probes: Inspiring Design for and with Families," *Proc. Conf. Human Factors in Computing Systems* (CHI 03), ACM Press, 2003, pp. 17–24.