Supporting Social Interaction with Smart Phones

The smart phone offers communication, connectivity, content consumption, and content creativity. Seven different systems exemplify its ability to support a wide range of social interactions, helping make pervasive computing a reality.

Russell Beale
University of Birmingham

The smart phone represents the current pinnacle of mobile phone development, coupling phone capabilities with the additional functionalities of a PDA. In this convergence between phone and handheld computer, the phone has the dominant genes—smart phones generally look more like phones than PDAs. The smart phone's evolution from the mobile phone influences how users tend to think of these devices, as reflected in the handset design. Smart phones are predominately communication devices, with additional computing power built in.

Compare this to PCs. Originally developed from larger computers that were pure calculating machines, PCs were initially desktop systems used for content creation and information storage. It was only with the advent of the Internet and browsers that they became strong communication tools. Now highly capable at communication, offering systems as diverse as email, Web publishing, instant messaging, and voice over IP, the combination of computing power and communication lets us envisage pervasive environments in which we can interact with and exploit the advantages of digital systems. But it's the advent of the smart phone that will let us realize some of these visions.

For smart phones to become successful pervasive system components, they must support and enhance various user activities and offer useful, effective functionality. Given their origins, clearly smart phones must support interaction and communication between people. In a pervasive environment, phones exist in a social setting where the focus is communication, not computation. Smart phones should still perform computations, store information, and support other typically computing-related tasks, but they're likely to be more successful if they do these things to augment communication.

This perception has guided the research of a group of students and I into the design of pervasive computing systems. The systems we've built focus on augmenting and enhancing communication; they're communication devices first and computers second. The systems I report on here support social interaction between people—in particular, interaction that would be difficult if not impossible to achieve without the smart phone technology. This dependency on the systems endows them with a degree of utility that drives their effectiveness, which is key to increasing the use of pervasive systems.

Supporting social interactions

The smart phone is a convenient, highly accessible, and capable device that's well suited to communication and yet can still create interesting content, whether it be video, audio, or text. It's a two-way device, creating and consuming information, is highly personal, and is almost always available, making it an ideal system for pervasive, supportive social computing.

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

Each of our smart phone applications examines one approach to augmenting social communication. We use a shorthand notation for interactions between people, loosely based on the number involved:

- 1 represents an individual, with 1–1 representing person-to-person communication
- *N* represents a group, with 1–*N* representing person-to-group communication and *N*–*N* representing within-group communication
- ∞ represents the world, with 1–∞ representing person-to-world communications and *N*–∞ representing group-to-world communication

If we can support all of these communication styles using the smart phone as the main user device, we can determine that it's indeed a useful device for pervasive computing, able to support a multitude of communication approaches. The collective effect is important; we can hardly call the smart phone a pervasive device if it only supports one form of interaction.

Localized dating service

To support interaction between individuals (1–1 interaction), a system should offer people a new experience or at least provide them with an easier solution to an old problem. To identify areas for development, we matched the smart phone system's features to interaction needs that conventional systems weren't adequately supporting. A smart phone's characteristics are easy to identify: they have screens with sufficient screen estate and resolution to show information (though nothing like that of notebooks or PCs), some processing power, some memory, and (free) short-range and (not free) long-range connections. Significantly, they're personal devices with private information stores, almost always used by one individual. They're text and image compatible, and because of their size and battery life, they offer permanent availability attached to the user.

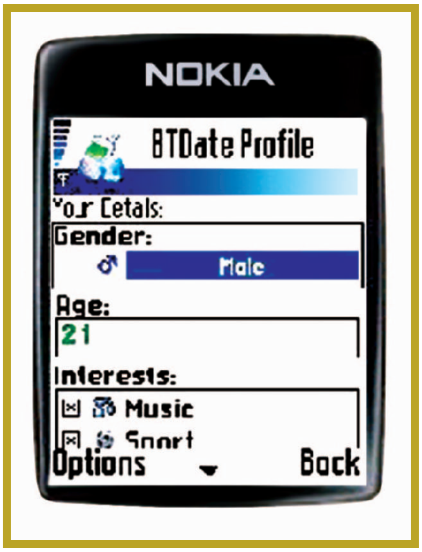


Figure 1. Bluedating profile interface showing the owner's gender, age, and interests.

Almost all smart phones support Bluetooth communication, a wireless standard offering free local (10–100 meter) connectivity. The Bluetooth protocol lets devices discover each other and exchange data with differing levels of user input. It's often used to communicate between the phone and a Bluetooth headset, allowing hands-free operation of the phone, or to exchange data between the phone and a user's PC.

Bluetooth technology maps well to dating: It requires people to be in close physical proximity and can be configured to allow information exchange without user input. Because smart phones are personal devices, they can contain personal and private information and can selectively share this information with other devices.

We call our dating application Bluedating. Users enter their interests and desires using the interface shown in Figure 1, as well as a profile of their desired partner. The system advertises this information (and only this information) over Bluetooth. A second part of the application continually searches for other profiles over Bluetooth. When it finds one, it compares the discovered profile to the desired profile. If the two profiles match, the system informs both users (usually by vibrating the phone) of the potential match. The rest is up to the users.

The system exploits the Bluetooth protocol's local nature, alerting the user to people nearby who are also looking for a date. This is quite different from Internet dating approaches, which use a similar profile system but might identify a potential partner who is many miles away. A localized system lets users see their matches without having to travel long distances or arrange a formal date and decide whether to pursue a relationship. It also lets people keep their personal information on their phone, unlike the Serendipity system, which has similar functionality but stores profiles on a central server and requires a phone connection to the server to compare profiles (see the related article in this issue, "Social Serendipity: Mobilizing Social Software"). Bluedating also shares some features with the profile-based Proem system[1] and LoveGety,[2] although both of these systems use hardware specifically designed for the dating application. Increasing the functionality of systems that users already possess brings us much closer to the pervasive computing ideal.

We ran trials of the Bluedating system at the University of Birmingham. We provided the system to everyone who wanted it and gave questionnaires and performed informal interviews with those who agreed to try it. Reactions were positive. Most users had initial reservations about such systems, as they did about similar online services, but found Bluedating more acceptable because it alerted them subtly, letting them interact as they chose. Because few users have worked with the system with any degree of seriousness, we can't as yet report any statistically significant results. However, the positive feedback encouraged us to redesign the Bluedating system to be more generic, thus letting us target a wider audience as we can develop additional applications.

Community building

The new system, called BT Communities, provides a software framework that can run and manage multiple Bluetooth services from a single application. This framework will let the device act as both client and server by offering services that remote devices can discover and connect to and by letting the device search for services offered by other devices. The framework provides a platform that can be easily extended with additional packages and new user groups offering their own functionalities.

To achieve this, we separated the application into three layers. The bottom layer—the *framework* layer—provides the basic shared functionality required throughout the application. This framework is completely independent of the subapplications built on top of it, letting us add new services simply by adding new subapplications. Integrating the new services into the rest of the application, however, requires another layer. This *integration* layer sits between the framework layer and the *subapplication* layer and is composed of a series of controller and processing classes. All of the modifications required to add a new service to the application occur in the integration layer.

The design of the BT Communities user interface follows the style outlined in the Nokia Series 60 UI Style Guide v1.0 (www.forum.nokia.com). BT Communities provides a simple menu-driven interface that lets users easily navigate through the application. Because a search can take a long time to complete, depending on the number of devices and services involved (a single device inquiry and service search typically takes around 11 seconds), the system provides status screens showing activity and cancel buttons. To keep the interface responsive at all times, the system performs lengthy or blocking operations, such as searches and running services, in a thread separate from the main system and user interface threads.

We built a joke-sharing application called JokeSwap on top of this framework, which lets people exchange jokes over Bluetooth. If someone has a joke in their joke store, the system offers it to other devices. Devices detecting the offer examine their joke stores for the joke, check their owners' personal profiles to see if it's the sort of joke they like, and, if so, accept it and offer a joke in exchange. This form of informal information exchange supports local community building. Mutual information sharing is part of the glue that binds a community together; hence, sharing jokes and other informal information is integral to supporting social systems.

We then rebuilt the Bluedating system on top of BT Communities. We also provided a chat facility, much like Instant Messenger, but local. The chat system has had extensive use, especially by students in lectures.

File sharing

Our relative success using Bluetooth to support individuals led us to extend the approach to groups, thereby supporting *N*–*N* interactions. We wanted to create a system that supports groups of people working together while still using the notions of localized connectivity. This aim is similar to that of Hummingbird,[3] but we wanted to more fully exploit smart phone capabilities and common uses. Many users store and share documents, data, videos, and photographs on their phones; we wanted our system to enhance this information sharing.

We thus created BT Share, which implements a peer-to-peer file-sharing system in which a user identifies a file-store on their phone as being public and open for sharing with the group. The system negotiates security protocols and locking mechanisms, letting other authorized users access information in the public store. Users can access, modify, and spread information (documents, music, and so on) among their group without needing a centralized server or explicit communication. As they pass each other in their everyday lives, their phones exchange and share relevant information.

> Many users store and share documents, data, videos, and photographs on their phones; we wanted our system to enhance this information sharing.

BT Share is similar to peer-to-peer systems common on PCs and the Internet (Kazaa, for example). Like BT Share, these systems offer a transparent simple sharing mechanism.

Because our system is Bluetooth-based, users must be near each other for the devices to exchange information. It therefore encourages face-to-face interaction. We've observed informally that the sharing mechanisms encourage users to share images and content that they would otherwise not explicitly send to other users. When users select an item, they provide a topic of conversation, thus enhancing interpersonal communication. By encouraging and reinforcing internal group relationships, BT Share achieves our goal of supporting groups and their interactions.

Shared spaces

We also designed two systems that support wider group communication: group to group (*N*–*N*) or, more generally, group to world (*N*–∞). We've noticed that in the coffee area outside our labs, where tables and chairs encourage people to sit and chat, students congregate and share information related to both work and wider social interests. Notice boards in the area collect much of the durable information. Our goal was to find more effective ways of supporting this sharing and exchange of public information.

Existing projects in this domain share ambient and specific information in public spaces[4–6] but don't focus on pervasive access. Mike Ananny and Carol Strohecker[7] present a public space into which users can text-message the captions or partial content of stories, but we focus on supporting typical conversations, rather than structuring conversations around artifacts.[8] We designed two systems that let users post messages to a shared space that we projected into the café area. This space is digital but has a clear expression in its projection into the public space, providing a central locus around which we expected interaction to occur. This lets users without smart phones or other devices view the information and participate in the interactions. This feature separates the approach from the BT Communities approach, which only connects technologically enabled participants. Users can post text or image messages from any mobile device.

We designed two systems that let users post messages to a shared space that we projected into the café area. This space is digital but has a clear expression in its projection into the public space.

The first system, PublicSpace, is based on a client-server architecture, with clients—smart phones, PDAs, and laptops—communicating via Short Message Service (SMS), Multimedia Messaging Service, 802.11b, or 802.11g technologies. Clients send messages to the Java-based server, which collates the messages into one display image and projects it.

The second system, SharedSpace, uses a peer-to-peer architecture, with each device in the system forming a node in the distributed digital space. Each node holds the same information. When new information is created, the system distributes it to the nodes, each one passing messages to the next (via SOAP[9] and HTTP). Users can also view the shared space on the Internet using some smart phones or a PC and can view a text version of the messages using a Wireless Markup Language interface for less powerful smart phones. Thus they can interact even when not physically colocated.

However, letting anyone post messages meant that material soon overwhelmed the available display space, with small groups of users sharing information that interested few others. In addition to posting new material, users wanted to comment on major postings. We therefore created an alternative shared space, public but organized by personal preferences. Our priority here was to address the clutter in the space as well as to create personalized views for small devices. Rather than provide a digital version of an existing solution, we took the conventional approach's good features and enhanced the user experience by applying the digital technology.

In the revised system, we analyzed each message as it was posted via the client and provided summaries of them on the display. We adopted a simple algorithm that removed common words from each message and subjected the remainder to a word frequency count. We then picked the most common words and identified a sentence or significant sentence fragment containing all of these words (or came as close to this as possible), using this as the summarization. We also used Bayesian statistics to classify messages by category and user interests, letting users develop personal profiles with the sorts of messages they most liked to see.

We provided two forms of display. A public view presented a news-oriented view of the data with summaries of the main stories appearing on the display projected into the communal space. Personal views on the shared data presented the same material, but ordered by user preference (based on the Bayesian analysis). So, some users would see news material as a priority, just as on the public display, whereas others would see sports first and then gossip. Depending on the client, the display would include more or less information. A laptop would show all of the content of the most recent messages and summaries of the older ones, whereas the smart phone client would show summaries only until the user chose to drill down into a story to read more or to comment.

We also used the Bayesian filter to automatically remove spam and extended the system by keeping the most active messages (those most frequently commented on or read in detail) near the top of the pile, rather than letting other postings displace them.

We also fed information from newsgroups and other digital sources into the system, providing a shared repository of material. Our intention was to provide the raw material for gossip and social exchange, even if the users contributed nothing initially themselves. These items—"coffee-room" or "watercooler" items—were to be the initial seeds for the informal conversations that typically take place in these locations.

Information only remained active in the space if people viewed and commented on it; uninteresting or stale material quickly disappeared from the displays. People enjoyed the shared and dynamic nature of the space. Our observations found that the space was used for different sorts of information exchange than the notice board that spawned the idea—rather than long-term notices, information moved through the system over the course of a couple of days, with a few items having greater longevity because they provoked comment after comment. The system became something of a cross between a bulletin board and instant messaging. However, by limiting the uploading of comments or new items to only when users were in the coffee area or its close vicinity, we retained a community feel.

Integrating situated interaction with mobile awareness

Our next system, the Intelligent Multimedia Messaging System (IMMS), addresses person-to-group (1–*N*) interactions.[10] A common problem in many learning organizations is the lack of any formal means to contact staff members rapidly without potentially disclosing personal information, such as mobile phone numbers. One traditional method of interaction between staff and students involves sticking notes on office doors. Lecturers wanting to leave messages on their office doors from remote locations must often rely on other staff members, such as a receptionist, to post a note on their behalf. Although this method has worked for a long time, there are intrinsic issues relating to a lack of security and privacy, and the practical problems of notes falling off doors or the lack of timeliness in posting the note.

To address these issues, IMMS uses mobile and other technologies to increase the interaction level between staff and students. We based the system on the concept of *situated interaction*[11,12] in an attempt to bridge the gap between learner and instructor and to increase mobility and remote accessibility, necessary for modern learning situations.

IMMS involves placing several display units (iPAQs or smart phones) on the office doors of various staff members to act as information and messaging terminals for students. Because these devices are placed at the locus of traditional interaction between participants, they are situated, and they afford communication possibilities because users arrive at the location with that express purpose in mind. A remote access Web-based management system lets the unit's owner set the display contents, typically a message or image. An SMS-based interface lets users update the display by sending a text message from their phone to the IMMS server. For example, staff members can update their display to inform scheduled visitors that they will be late because they're stuck in traffic.

Student members of the department can not only view the image and textual message but also send messages to the owner via the display unit interface. A student coming by to see a lecturer and finding an empty office can use the display unit to compose a short message. IMMS will either send the message via text message to the owner's mobile phone or store the message in the management system for display the next time the IMMS user logs into the system, using heuristics to determine which is appropriate. This intelligent routing of messages extends similar work on situated door displays by Keith Cheverst and colleagues,[13] which used unintelligent SMS notification. The system's Web-based components offer configuration and message management and also allow remote access to the screen display, so users on the Internet can find lecturers' statuses without having to go to their doors.

We evaluated the system with the help of six students representing a cross-section of potential users. Using a scale from 0 (bad) to 10 (good), we asked them to score the interface's look and usability (9.33, standard deviation 0.67), the display's content and appropriateness (7.83, s.d. 0.94), and the system's usefulness and functionality (9.00, s.d. 0.42). The results indicate that the users saw value in the system, which clearly met our goals in providing a pervasive and appropriate way for students and staff to communicate with each other.

Mobile blogging

Our next system, SmartBlog, used mobile blogging to support one-to-world interactions (1–∞). Mobile devices in general, and smart phones in particular, have expanded beyond their communication origins to become accomplished photo- and video-creation tools, able to provide

Mobile devices in general, and smart phones in particular, have expanded beyond their communication origins to become accomplished photo- and video-creation tools.

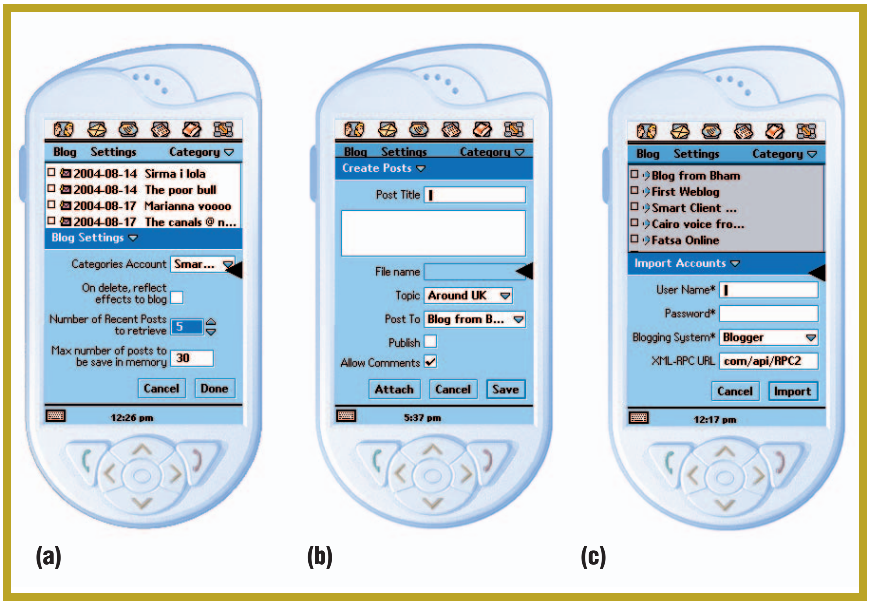


Figure 2. Three examples of user input dialogs: (a) define settings, (b) create posts, and (c) import accounts.

immediate images and text regardless of the user's location. *Blogging* is the process of publishing a personal diary or journal online. The resulting *weblogs*, or *blogs*, are simple layouts of personal posts ordered chronologically. Mobile blogging makes this process even more immediate, and given that we like pictures and photographs, the ability to post multimedia content makes a mobile blogging client desirable. A mobile blogging tool could be the best shortcut between bloggers and their weblogs if it overcomes the issues mobile applications face—for example, unreliable and interrupted network signals, limited interface capabilities, and limited processing power.

Existing mobile blogging clients tend to make trade-offs that limit their usefulness. For example, Azure (http://web.vee.net/projects/azure), a Java-based blogging client, prioritizes targeting a wide set of smart phones over interface usability, and doesn't provide multimedia publishing. On the other hand, Nokia's LifeBlog application (www.nokia.com/nokia/0,1522,,00.html?orig=/lifeblog) provides a slick interface with some interesting photo-editing and synchronization options on a subset of Nokia phones. However, LifeBlog doesn't support instant online publishing; instead, it creates a local diary or gallery of annotated photos on the user's PC, which can then be uploaded—not much use if you're away from the PC for a long time.

We interviewed some users to determine what features we needed to provide. All interviewees were active bloggers, and all felt that a decent mobile blogging system should be more than a way to publish camera phone images on a Web site. Blogs are fairly immediate, thus requiring more frequent revising than crafted Web sites, so editing and management abilities are important. The users also all wanted to use multimedia if it were simple, but 67 percent were concerned about the potential costs.

We therefore designed and implemented a client that targets the Symbian-based smart phone series. SmartBlog provides an easy remote-management tool for blogs because it's designed to communicate with several blogging accounts (possibly heterogeneous) at the same time. SmartBlog offers all the regular blogging options for retrieving, categorizing, publishing, and editing blog posts. In addition, it provides automated multimedia publishing. SmartBlog uses HTTP, communicating as a browser does; therefore, it works over any type of Internet connectivity. The SmartBlog architecture is multithreaded, letting the smart phone function as a mobile phone without compromising its behavior or performance. No matter SmartBlog's state, the phone should continue to receive and place calls.

The SmartBlog input dialog screenshots in Figure 2 demonstrate the system's ability to manage different accounts and create and post information with minimal effort. Typical blog postings take between 20 and 50 seconds, depending on the image size and amount of text. One advantage of SmartBlog is that it customizes the multimedia produced from the phone to fit the limits imposed by publishing systems. We designed SmartBlog to reduce captured images to sensible sizes (all customizable) for Internet display (see Figure 3), thereby saving connection time when publishing and download time for the users' viewing.

Six users tested SmartBlog, and they found it fun and easy to use, leading them to post more images to their blogs than usual. No one reported problems with installation or the interface, though most had problems setting up Bluetooth connectivity (a known problem with Nokia's PC Suite router). Regardless, 67 percent favored Bluetooth because it's free and fast. Moreover, all of the users were satisfied with the system's performance, especially because it didn't block any other phone functionality.

Pervasive computing will become a reality when systems provide appropriate support and let us achieve our desires in new and effective ways. Each of the systems we've described has demonstrated that smart phones support interactions between users, whether one to one or many to many. More importantly, the smart phone platform supports all seven of our applications.

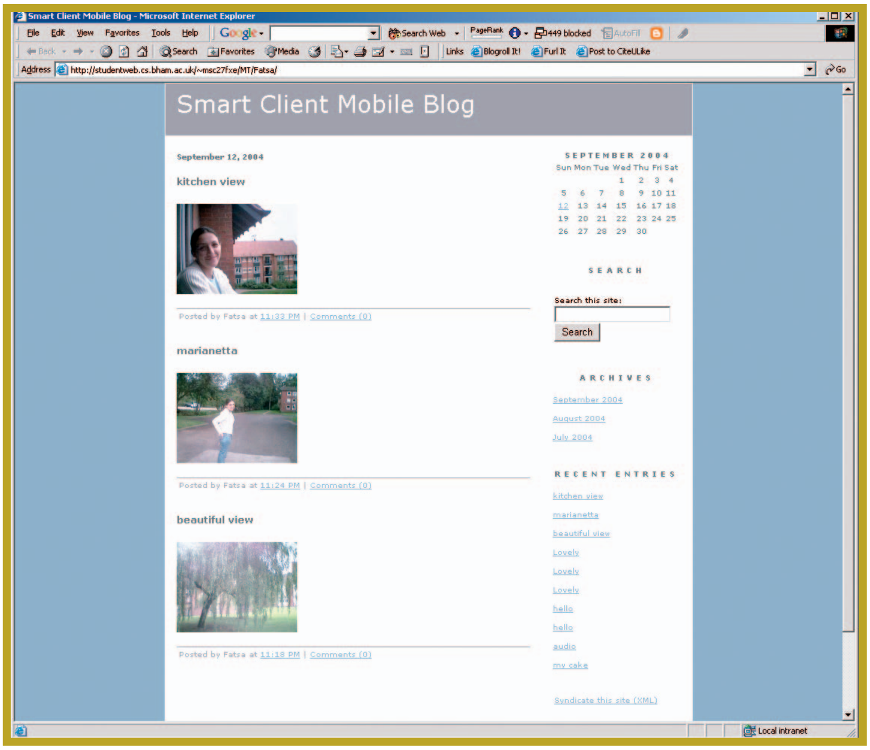


Figure 3. A blog produced with SmartBlog. The system configures images to appear this size automatically.

The smart phone's ability to support so many enhanced forms of communication suggests that it's both pervasive and flexible. Furthermore, as our research has demonstrated, it can improve users' sociability. By using the device's technical characteristics, we can produce designs that exploit the infrastructure and can thus develop new approaches to supporting interpersonal communication. ■

ACKNOWLEDGMENTS

I thank the students and colleagues who assisted in designing, building, and evaluating the concepts and applications described here: Nick Stoppard, Fatma Elsayed Meawad, Matthew Jones, and Nickolas Pitsioras, and the many unknown but invaluable users and students who participated in trials. I also thank the reviewers for their suggestions and comments.

REFERENCES

1. G. Kortuem, Z. Segall, and T.G.C. Thompson, "Close Encounters: Supporting Mobile Collaboration through Interchange of User Profiles," *Proc. 1st Int'l Symp. Handheld and Ubiquitous Computing* (HUC99), H.-W. Gellersen, ed., LNCS 1707, Springer-Verlag, 1999, pp. 171–185.

2. Y. Iwatani, "Love: Japanese Style," *Wired News*, 11 June 1998; www.wired.com/news/culture/0,1284,12899,00.html.

3. L. Holmquist, J. Falk, and J. Wigstroem, "Supporting Group Collaboration with Inter-Personal Awareness Devices," *J. Personal Technologies*, vol. 3, no. 1–2, 1999, pp. 105–124.

4. D. Vogel and R. Balakrishnan, "Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users," *Proc. 17th Ann. ACM Symp. User Interface Software and Technology*, ACM Press, 2004, pp. 137–146.

5. S. Izadi et al., "Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media," *Proc. 16th Ann. ACM Symp. User Interface Software and Technology*, ACM Press, 2003, pp. 159–168.

6. S.K. Card, G.G. Robertson, and J.D. Mackinlay, "The Information Visualizer, an Information Workspace," *Proc. ACM Conf. Human Factors in Computing Systems* (CHI 91), ACM Press, 1991, pp. 181–188.

7. M. Ananny and C. Strohecker, "Designing Public Spaces for Democratic Stories," *Proc. 1st ACM Workshop on Story Representation, Mechanism, and Context*, ACM Press, 2004, pp. 47–50.

8. G.W. Fitzmaurice, "Situated Information Spaces and Spatially Aware Palmtop Computers," *Comm. ACM*, vol. 36, no. 7, 1993, pp. 39–49.

9. M. Gudgin et al., eds., "SOAP Version 1.2 Part 1: Messaging Framework," World Wide Web Consortium, 2003, www.w3.org/TR/soap12-part1.

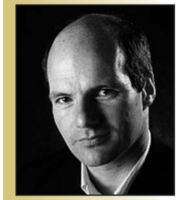
10. R. Beale and M. Jones, "Integrating Situated Interaction with Mobile Awareness," *Proc. 3rd European Conf. Mobile Learning* (MLEARN), E. Murelli, G. Da Bormida, and C. Alborghetti, eds., Center for Research on the Applications of Telematics to Organizations and Society (CRATOS), 2004.

11. M. Hirakawa and K.P. Hewagamage, "Situated Computing: A Paradigm for the Mobile User-Interaction with Multimedia Sources," *Ann. Software Eng.*, vol. 12, no. 1, 2001, pp. 213–239.

12. N.A. Streitz et al., "Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments," *Proc. 10th Int'l Conf. Human-Computer Interaction* (HCI Int'l 2003), Lawrence Erlbaum Assoc., 2003.

13. K. Cheverst et al., "Exploring Situated Interaction with Ubiquitous Office Door Display," *Proc. Int'l Workshop Situated Interaction, Conf. Computer-Supported Cooperative Work* (CSCW 02), ACM Press, 2002.

the AUTHOR



Russell Beale leads the Advanced Interaction Group in the School of Computer Science at the University of Birmingham, UK. His research interests include supporting social interaction, mobile and pervasive systems interaction, creative design, and information visualization and access. He received his DPhil in computer science from the University of York. He's chair of the British Computer Society's HCI Special Interest Group. Contact him at Advanced Interaction Group, School of Computer Science, Univ. of Birmingham, Edgbaston, Birmingham B15 2TT UK; r.beale@cs.bham.ac.uk.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.