A Design Space Analysis of Availability-Sharing Systems

Juan David Hincapié-Ramos
IT University of Copenhagen
Copenhagen DK-2300 Denmark
jdhr@itu.dk

Stephen Voida, Gloria Mark
University of California, Irvine
Irvine, CA 92697–3440 USA
{svoida, gmark}@uci.edu

ABSTRACT

Workplace collaboration often requires interruptions, which can happen at inopportune times. Designing a successful availability-sharing system requires finding the right balance to optimize the benefits and reduce costs for both the interrupter and interruptee. In this paper, we examine the design space of availability-sharing systems and identify six relevant design dimensions: abstraction, presentation, information delivery, symmetry, obtrusiveness and temporal gradient. We describe these dimensions in terms of the tensions between interrupters and interruptees revealed in previous studies of workplace collaboration and deployments of awareness systems. As a demonstration of the utility of our design space, we introduce InterruptMe, a novel availability-sharing system that represents a previously unexplored point in the design space and that balances the tensions between interrupters and interruptees. InterruptMe differs from previous systems in that it displays availability information only when needed by monitoring implicit inputs from the system's users, implements a traceable asymmetry structure, and introduces the notion of per-communications channel availability.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces—graphical user interfaces.

General terms: Design, Human Factors

Keywords: Workplace awareness, availability, interruptibility

INTRODUCTION

Effective collaboration in the workplace requires various modes of interaction. Co-located and remote colleagues may interact face-to-face or over different media such as the telephone, instant messaging, and email. Above all, they must maintain a general awareness of one another. Researchers have explored multiple ways of providing support for this kind of awareness through a class of computational systems that are generally characterized as *awareness systems*. Awareness systems are motivated by the observation that workplace awareness leads to more interactions among colleagues (both remote and collocated) and a general improvement in performance [30]. However, while increased collaboration is often considered beneficial, it also incurs costs: people can interrupt their collaborators at inappropriate times, leading to increased task switching, redundancy in work, and stress [27, 28].

Guided by the premise that better timing of interruptions decreases their disruptiveness [5], a subset of awareness systems focus on communicating the *availability* of others with whom one collaborates. The goal of these systems is to help collaborators identify the most appropriate (and least costly) times to initiate interactions. Availability-sharing systems provide this kind of awareness by directly communicating the other person's activity (e.g., high-fidelity video and audio links [9, 40] or raw sensor values [12]), generating and sharing a more abstract representation of a person's interruptibility drawn from multiple cues (e.g., sensor data [25]), and/or by providing a visualization of historical patterns of activity.

The goal of our research is to examine the design space of availability-sharing systems in terms of the costs and benefits for collaborating partners and to apply this insight to a system design that can optimize the trade-offs. Designing a successful availability-sharing system requires striking the right balance between maximizing the benefit for, and minimizing the costs incurred by, each of the participants in a collaboration (Figure 1). For the *interrupter*—the person who is initiating the interaction—the system should facilitate making a quick, accurate decision about whether, when, and through what communications medium it is appropriate to interrupt the other person. Prior studies of awareness systems suggest that it is beneficial to share more than simple presence information to inform this decision-making process [16], and that it is essential to provide relevant information in a timely fashion [35, 40, 41, 43]. Some of the costs to the interrupter that should be minimized include the effort required to access the availability information, the cognitive overhead needed to correctly

Figure 1: Availability-sharing systems provide information to a potential *interrupter* that facilitates decision making about whether to interrupt a colleague (the *interruptee*), when to do so, and what communications medium to use.

The definitive version was published in *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '11)*, Santa Barbara, California, October 16–19, pp. 85–96.
http://doi.acm.org/10.1145/2047196.2047207

interpret the availability representation(s) [3, 12], and the obtrusiveness of the availability information display [25]. In contrast, the main cost to be minimized for the *interruptee* is the disruptiveness of the communication, which can be achieved by suggesting a communications medium that fits their current activity. Other important costs, such as privacy (i.e., the fidelity of the data [9, 40] and with whom those data are shared [25]), must also be considered.

In this paper, we examine the design space of availability-sharing systems, drawing attention to the inherent design tensions and the pragmatic challenges that others have encountered in attempting to foster collaboration and reduce the intrusiveness of workplace interactions. Based on a review of existing availability-sharing systems, we provide a framework for understanding the different design trade-offs in sharing availability information, highlighting the ways that different implementation and interface decisions provide different degrees of cost and benefit to the participants in these systems. To illustrate how this framework might guide the design of novel awareness systems, we present one example of a system designed specifically to balance the needs of the interrupter and the interruptee.

This research contributes to the literature on availability-sharing systems in three primary ways:

- We present a design space for availability-sharing systems, grounded both in prior theoretical and practical work in this area;
- We discuss how different points in this design space offer different levels of cost and benefit to the different parties in a collaboration; and
- We present the design of InterruptMe, an availability-sharing system that exemplifies an unexplored point in the design space, balancing the needs of interrupters and interruptees, and suggests new directions for research and development in this domain (Figure 2).

RELATED WORK

In this section, we examine the research literature related to the evolution of the notion of availability.

Early Explorations of Availability

Media spaces connect remote locations through audio and video links [9, 40] or slowly updated images [6], aiming to provide remote colleagues with the degree of awareness about presence or current activities that is available to co-located colleagues. One of the key roles for these systems was to enable participants to estimate the availability of remote colleagues before initiating interaction. Fogarty et al. [11] questioned whether video and audio links were good indicators of interruptibility. They carried out a study of office workers' capacity to estimate the availability of another person from snippets of video. Their study revealed that humans are only slightly better than chance in estimating a person's self-reported availability on a 5-point scale.

These explorations illustrate several of the challenges faced by availability-sharing systems that are based on the model of continuously transmitting audio and video streams among collaborators. Video and audio links may not provide sufficient information to a potential interrupter to determine their colleagues' availability. Additionally, sharing live video or audio necessarily incurs privacy costs for the interruptee. Finally, displaying the media space video feeds often requires an auxiliary display device or a large dedicated window region on the interrupter's computer, which can be both obtrusive and potentially distracting.

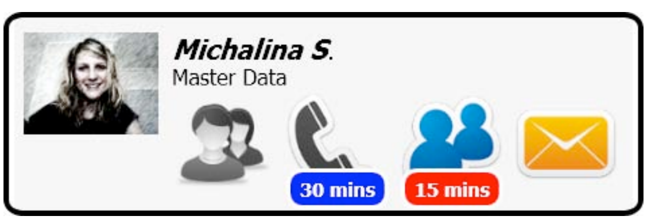

Figure 2: InterruptMe shows different availability values for each communication medium. Michalina is unavailable for face-to-face and phone interactions (monochrome icons), but is available by instant messaging and email (full-color icons). The system can also provide historical context; in this case, Michalina has been available via IM for 15 minutes, but away from her phone for 30 minutes.

Measuring Availability

The Peepholes [14] system represented a different approach from previous video-based systems, providing awareness without incurring the hardware costs and privacy concerns of media spaces. It provided information about its users' availability by using iconic presence indicators. Peepholes introduced the idea of modeling availability based on users' network connectivity and computer activity levels. This model determined whether a user was working, idle, not logged in, or in an unknown state. The main limitation of this system was that presence, taken alone, communicates little about how busy, and, therefore, how interruptible a person is. Hudson et al. [19] took this approach a step further, and by studying the disruptive effect of interruptions in collaborative work environments suggested modeling personal availability as a function of *interruptibility*.

Different research has measured interruptibility through the use of both manual input and automatic data collection. Manual input was studied in the live addressbook project [35], where users manually entered availability information, which was then distributed to their contacts. A deployment of this system revealed that the costs of maintaining status messages and availability states were unreasonably high, potentially undermining the value of such systems. Automatic data collection has been studied in several systems, such as BlueSpace [24], ConNexus [41], MyVine [12], and SideShow [4]. These systems have all used a combination of computer activity, location, calendar information, and manual user input to model a user's availability. Lilsys [2] and the attentive cellphone [43] used sensor data to infer interruptibility and share availability information with one's colleagues. This approach was generally well received, with some of the studies lasting several months and users continuing to use the system afterwards. However, several problems were identified over the course of these studies. Interruptees reported privacy concerns due to the number of sensors deployed in their office spaces. Interrupters often misinterpreted awareness cues because it was difficult to interpret availability representations; they also often ignored the system due to the high interaction costs of accessing the availability information.

Modeling Availability

Most systems model availability as a single value, often through an icon. Some systems (e.g., MyTeam [24] and MyVine [12]) support gradual refinement of availability information as the user clicks on (or "drills into") specific contacts. These refinements include the raw values collected by sensors deployed in the workspace, a recommended communication channel, the expected time of return, or personal contact information. Other projects, like BlueSpace [24] and Labradoor [36], explored moving availability information off the desktop and into the environment. In doing so, these systems reclaim some of the display real estate required by always-on applications, but raise additional questions about the distractions of desktop and semi-public sharing of availability information.

Some research has questioned the assumption that availability information is one single and comprehensive measure. Horvitz and Apacible [17] investigated how the disruptiveness of an interruption depends on the communication channel, and found considerable differences between different media, such as telephones, pagers, and desktop appliances. In another study, Harr and Wiberg [16] found that *being busy* does not mean *being unavailable* for communications; being busy changes the likelihood of using a particular communication channel and the expected response. They go further to say that availability is also affected by the relationship between the communication partners. Voida et al. [44] and Patil and Kobsa [38] found that users have varying availability preferences, as demonstrated by having different IM accounts for work and for friends.

Gross and Oemig [15] developed the PRIMIFaces system, allowing users to maintain *selective availability*, that is, different indications of availability for different groups of contacts. Users reported that this approach requires a considerable manual effort to keep all of the various status indications up-to-date. In a subsequent study, Fetter et al. [8] instrumented PRIMIFaces to capture sensor data, and performed an experience-sampling experiment to determine the feasibility of *lightweight* selective availability. Their results show that users considered selective availability important, that general availability consistently differs from the one reported in the IM client, and that selective availability strongly relates to the users' locations.

DESIGN SPACE FOR AVAILABILITY-SHARING SYSTEMS

In order to better understand the trade-offs inherent in the design of availability-sharing systems, we undertook an extensive literature review of prior research on interruptions and of systems that communicate availability information among a group of colleagues. In this section, we construct a design space that articulates the axes serving to distinguish among the capabilities of these systems. Our design space seeks to foster research in three important ways: (1) to identify previously unexplored points in the design space, for designing new systems; (2) to introduce techniques that represent new values for a given dimension, as we do later on with traceable asymmetry; and (3) to propose new ways to realize an existing point along an axis that clearly depart from approaches taken in prior systems.

We took a two-stage approach to define our design space for availability-sharing systems. Our first activity was a **top-down review** of the literature aimed at understanding existing taxonomies in the domain of human interruptions. From this activity, we identified MacFarlane's work as the most comprehensive. McFarlane [33] defined a taxonomy of human interruptions with the following dimensions: (1) source of interruption, (2) individual characteristics of a person receiving interruption, (3) method of coordination, (4) meaning of interruption, (5) method of expression, (6) channel of conveyance, (7) human activity changed by interruption, and (8) effect of interruption. McFarlane and Latorella [34] described this taxonomy as an attempt to map the whole space of interruptions and to determine the areas where technology interventions might be helpful.

MacFarlane's *source of interruption* dimension points out places for technological intervention, such as technologies to better deliver system messages to a user (see Jonsson et al. [23] on notification systems) and applications to help users better time their interruptions (availability-sharing systems). However, their general taxonomy provides little guidance for developing each particular kind of system, and especially for availability-sharing systems, it lacks direction on how to help an interrupter make decisions about how and when to initiate interruptions.

Our second activity consisted of a **bottom-up analysis** of existing design spaces in the related areas of *awareness systems*, *ambient displays*, and *notification systems* [13, 29, 31, 39, 42]. In contrast to the theoretically driven models of interruption provided by our top-down analysis, we wanted to understand how these alternative design spaces, derived more from the experience of *building* and *deploying* systems, overlap and extend the design space of availability-sharing systems. Even though availability-sharing systems have a different purpose than these broader classes of systems (as previously discussed in relation to notification systems), the issues involved in designing them are also often present when designing availability-sharing systems.

Combining all of these efforts, we created a more comprehensive set of design space dimensions that relate to availability-sharing systems, identifying resonances among the individual design spaces and collapsing similar axes together. In our final design space, we elected to focus on those aspects of the design that most clearly illustrate design opportunities that affect the cost/benefit trade-offs for the interrupter, the interruptee, or both parties. Our resulting design space, shown in Table 1, is derived from the

Design Dimensions	Types of Values			
Abstraction	Sensor data (e.g., motion sensor values) [2, 3, 41[1]]	Availability (e.g., manual, measured, or predicted) [1, 4, 12, 18, 21, 22, 24[1], 24[2], 24[3], 26, 35, 41[2], 43]	Natural (e.g., audio, video) [6, 9, 40]	Multimedia (e.g., status message annotating a video feed) [32, 36]
Presentation	Continuous (e.g., gradual fade) [12, 36]	Discrete (e.g., icons) [2, 3, 4, 14, 18, 21, 22, 24[1], 24[2], 24[3], 26, 41[1], 41[2], 43]		Literal (e.g., video) [1, 6, 9, 32, 35, 40]
Information Delivery	Always on (e.g., dedicated display) [4, 9, 14, 24[3], 26, 32]	Almost always on (e.g., overlapping window) [2, 3, 6, 12, 22, 21, 41]	On request (e.g., query-based system) [1, 18, 35, 40, 41[2], 43]	Implicit interaction (e.g., situated display) [24[1], 24[2], 36]
Symmetry	Symmetric, Traceable (e.g., contact-based with request history) [9, 18, 22, 40]	Symmetric, Blind (e.g., contact-based) [1, 2, 3, 4, 6, 12, 21, 24[3], 32, 41[1], 41[2], 43]	Asymmetric, Traceable *(none)*	Asymmetric, Blind (e.g., public display) [14, 24[1], 24[2], 26, 35, 36]
Obtrusiveness	Focal [1, 14, 22, 32, 35, 40, 43]	Selectively focal [2, 3, 4, 6, 12, 18, 21, 41[1], 41[2]]	Secondary appliance [9, 24[3], 26]	Peripheral [24[1], 24[2], 36]
Temporal Gradient	Historical Availability *(none)*	Recent Availability [2, 18, 41[1], 41[2]]	Current Availability [1, 3, 4, 6, 9, 12, 14, 21, 22, 24[1], 24[2], 24[3], 26, 32, 35, 36, 40, 43]	Predicted Availability *(none)*

Table 1: Design space dimensions for availability-sharing systems and the position on each axis occupied by 22 existing awareness and notification systems. For systems that could occupy more than one cell on an axis, we categorized them into the cell in which the system would most frequently be used. [24][1] is BlueSpace's office front display, [24][2] is BlueSpace's LED ambient lamp, and [24][3] is the MyTeam component; [41][1] is ConNexus and [41][2] is Awarenex.

concerns of interruptions research, the experiences of systems design, and with the unique constraints and trade-offs of availability-sharing systems in mind.

Although grounded in the taxonomies presented by MacFarlane et al. [33], Matthews et al. [31], and others, our design space differs in multiple ways. MacFarlane et al. aim to map the whole space of interruptions, including non-technical issues like the effect of the interruption, and they attempt to suggest interventions to minimize the impact of interruptions, such as (general) availability sharing, availability modeling, notification delivery systems, etc. In contrast, our design space is specifically geared towards informing technical support in the design of availability-sharing systems. Second, whereas McFarlane et al. focus on the interruptee and Matthews et al. focus on the interrupter (they talk about the "receiver of the information"), our work considers the involvement and interplay of both.

In order to characterize the different values that exist along each of the design space axes, we reviewed the systems-oriented literature and collaboratively coded a wide variety of awareness systems according to each of our dimensions (see Table 1). Based on this classification process, we can more clearly explain the design possibilities along each axis and begin to identify positions in the design space and combinations of system features that are underexplored and may warrant further consideration.

Abstraction

At what granularity (or level of abstraction) are availability data collected and shared with one's colleagues?

A system's positioning along the abstraction axis relates to the types of data that are collected from the interruptee and used to model availability and/or are shared with colleagues. These data are used to calculate how interruptible a person might be, information that can be used by an interrupter to determine whether to carry out an interruption and when and how to do so. A system using *sensor data* abstractions requires interrupters to interpret low-level data directly (e.g., speech level, presence at desk) to determine interruptibility. MyTeam's conversation indicator is an example of this approach [25]. Other systems model *availability* more holistically by aggregating a variety of sensor values. Such systems assume that users can more quickly and unambiguously understand an interpreted measure. Examples of systems adopting this approach include Awarenex [41], BlueSpace's door light system [24], Lilsys [2], the live addressbook [35], and MyVine (at the first level of detail) [12]. A system that presents *natural* cues, like audio and video, relies on the human capacity to read such cues, to determine the interruptibility level, and to identify the best way to interrupt. This is the case in media spaces like Cruiser [9] and Montage [40]. Finally, a system presenting *multimedia* accounts of availability relies on users to read text, images, or videos, and apply contextual knowledge of the situation. Such is the case for online collaborative environments like the one studied by Harr and Wiberg [16] or public embedded displays like the Labradoor [36].

Dimension Values: Sensor data, Availability (manual, measured, predicted), Natural (audio, video), Multimedia.

Presentation

How are availability data presented to the system's users?

In our design space, we associate presentation with Matthews' concept of *abstraction level* and borrow her definition (we use a different name to draw a stronger distinction between this output-related dimension and the previous, input-oriented one):

> Abstraction involves extracting features or reducing the fidelity of information so that it is easier to read "at a glance" than the raw input. Abstraction enables lower attention consumption of information. One reason for this may be that abstract (pictorial) displays aid recall when compared to literal (textual) displays [31].

When no abstraction is performed on the presented data, this dimension takes the value of *literal*. Such is the case for camera-based or text-based systems like media spaces [9, 40], the AwarePhone [1], and the live addressbook [35]. Availability information presented as *discrete* values is the most common approach, and can be seen in systems like ConNexus [41] and Lilsys [2], where availability is shown as a binary value. *Continuous* values can be seen in IMBuddy [18] and MyVine [12], where availability is shown in a range from 0 to 1 (or as a percentage).

Dimension Values: Continuous, Discrete (e.g., icons), Literal.

Information Delivery

When are availability data presented to the system's users?

An *always-on* system constantly presents availability information to the user, minimizing the acquisition cost but potentially creating visual overload. Examples of always-on systems include the dedicated screens of the early media spaces or screen widgets like SideShow [4]. An *almost–always-on* system also continually presents colleagues' availability information, but can be moved into the foreground or background as needed, as in desktop applications like MyVine [12] and Piazza [22]. An *on-request* system has relatively higher overhead, as users must query the system and wait for a response each time, as in AwarePhone [1]. A system designed around *implicit interaction* presents information whenever the user needs it, as he or she moves around the environment or interacts with communication systems. Such systems pose a minimal acquisition cost and create no visual overload, as demonstrated in some components of BlueSpace [24].

Dimension Values: Always on, Almost-always on (e.g., overlapping window), On request, Implicit interaction.

Symmetry

To what extent does the system require or enforce reciprocity in the exchange of availability data? To what extent does the system provide traceability, or accountability, for the dissemination of this information?

Symmetry refers to the broadcast of availability information of users such that it generates mutual knowledge of one another [45]. Symmetry can also be seen as the level of social translucency supported by the system [7]. A *symmetric* system will only share availability data with those individuals from whom availability data is being received. Such is the case in most contact-based systems like MyUnity [3] and MyVine [12]. *Asymmetric* systems do not require disclosure of one's own availability in order to access others' availability information. Such is the case of public displays like BlueSpace's LED door component [24] and Peepholes [14]. The *traceability* sub-dimension characterizes the extent to which users know how others see them, that is, when someone has looked at their availability information and what they saw. This distinction is similar to the idea of accountability in socially translucent systems [7]. Most existing availability-sharing systems are *blind* and do not provide traceability. There are a few notable exceptions—for example, in Cruiser [9], users know when they are being observed because the system automatically establishes a two-way video channel. Traceability can also be subtle, as in IMBuddy [18], where the system accumulates a history of availability requests from other users.

Dimension Values: Symmetric (traceable), Symmetric (blind), Asymmetric (traceable), Asymmetric (blind).

Obtrusiveness

To what degree is availability data presented in the focus?

This dimension relates to whether the availability information is delivered in the focus or the periphery of the interrupter's attention (after [10]). Systems that obtrusively embed their output into the primary display surface (i.e., the computer desktop) make it easy to maintain awareness about others' availability. These systems also emphasize collaboration by prominently displaying the information gleaned from team members. This is the case for desktop collaboration systems like CommunityBar [32] and Piazza [22], for example. Colleagues' availability information can be moved incrementally towards the interrupter's periphery, such that it is made focal only when needed (*selectively focal*), or off-loaded entirely onto a *secondary appliance* or a *peripheral display*. The further into the periphery the information is pushed, the less it interferes with primary work tasks, but the more overhead is required to incorporate availability information into the center of attention when it becomes relevant or is needed.

Dimension Values: Focal, Selectively focal, Secondary appliance, Peripheral.

Temporal Gradient

On what time scale are availability data communicated?

This dimension relates to the use of temporal modifiers on the availability information presented to an interrupter. Temporal modifiers can present historic, recent, current, and/or predictive accounts of availability. *Historic* modifiers represent a sensor value or an availability measure in relation to large-scale previous values (e.g. the last month, days like today). *Recent* modifiers present it in relation to near-range previous values (e.g. the last hour). A *current* modifier presents the actual value. Finally, a *predictive* modifier uses information about past and current availability to estimate future interruptibility. One example of how predictive modifiers might be realized is a machine-generated indication of how long an interruptee's current availability state is likely to last.

Even though a temporal gradient has been previously used in awareness systems, as documented by Tomitsch [5], it has not been widely used in availability-sharing applications. The only exceptions are the *recent* modifiers used by IMBuddy [18], Lilsys [2], ConNexus, and Awarenex [41].

Dimension Values: Historical, Recent, Current, and Predicted availability.

DESIGN GOALS FOR AVAILABILITY-SHARING SYSTEMS

The central tenet of our research is that designing an effective availability-sharing system is primarily a matter of balancing the costs and benefits for each of the system's users. However, accomplishing this goal is not always straightforward, as some design decisions result in a preferable cost/benefit trade-off for one party at the expense of the other. Based on our analysis, we have derived a high-level overview of which positions along each design axis result in the most favorable cost/benefit value for the interrupter, and likewise for the interruptee (Figure 3). These considerations are the result of our analysis of the motivations for and discussions of the implementation and deployment of various availability-sharing systems from the research literature. We paid particular attention to instances where researchers described the shortcomings of their design decisions, feedback from users, and why (in some cases) users stopped using the system. For both the interrupter and interruptee, we found that there are some dimensions for which we could recommend a value; that there are some dimensions for which there is no clear tendency toward any single value; and that there are some dimensions that appear less relevant for a particular role.

Optimizing for the Interrupter

For the interrupter, previous research points toward using the "availability" value along the *abstraction* axis, because sensor data, natural and multimedia values all present several drawbacks. One of the concerns expressed in prior studies is the misinterpretation of information shared by remote users. For example, MyVine [12] assumed speech as an indication of a person being unavailable. However, users interpreted the presence of speech as an indication that the user was present and not engaged in a task demanding quiet attention, thus rendering them available for interaction. This suggests that using higher-level abstractions in availability-sharing systems might be more beneficial than sharing low-level or raw data. Similarly, natural and multimedia abstractions (audio & video) lead to increased mis-

Dimensions	Optimal Values—Interrupter			
Abstraction	Sensor	Avail.	Natural	MM
Presentation	Cont.	Discrete		Literal
Information Delivery	Always	Almost	Request	Implicit
Symmetry	S(T)	S(B)	A(T)	A(B)
Obtrusiveness	Focal	Selective	Appliance	Periph.
Temporal Gradient	Hist.	Recent	Current	Predict.

Dimensions	Optimal Values—Interruptee			
Abstraction	Sensor	Avail.	Natural	MM
Presentation	Cont.	Discrete		Literal
Information Delivery	Always	Almost	Request	Implicit
Symmetry	S(T)	S(B)	A(T)	A(B)
Obtrusiveness	Focal	Selective	Appliance	Periph.
Temporal Gradient	Hist.	Recent	Current	Predict.

Figure 3: The optimal values for each design dimension differ for interrupters and interruptees. (Shaded cells represent more preferable design solutions than un-shaded cells.)

interpretation of colleagues' interruptibility, as discussed by Hudson et al. [19].

The *presentation* used to display availability data to the interrupter impacts the cognitive overhead required to assess the interruptee's availability and the accuracy of these inferences. When literal data are presented to an interrupter, it is up to the interrupter to make sense of the data and draw their own inferences about whether or not it is an appropriate moment to interrupt. At the other extreme, highly abstract presentations of the data can provide a meaningful representation at-a-glance, yet might remove context that can be important in distinguishing nuance. The optimal choice might be a middle ground—clear and relatively unambiguous discrete presentations that make it easier to assess availability at-a-glance, with the ability to "drill down" into more literal data when needed.

Another concern from the interrupter's perspective is that systems often fail at delivering availability information at the appropriate time. Windowed availability-sharing applications have to be explicitly called to focus before they can inform interactions. At the other end of the spectrum, always-visible displays can be distracting and can suffer from the effects of change blindness [20]. The same set of sensors used to determine a user's availability could be used to provide the timely *delivery of information* when a user acts as interrupter by detecting implicit interaction. Similar to the service provided by augmented environments (e.g. BlueSpace [24] and Labradoor [36]) in communicating the presence of an office occupant to potential visitors, availability-sharing systems could provide information about a colleague's availability at the moment that the interrupter begins to compose an email or picks up a telephone handset. As a result, availability information would be delivered in a timely fashion (after the user decided to interrupt, but before the interruption actually occurred) and without incurring the extra cognitive load of explicitly querying the availability-sharing system.

The dimension of *symmetry* defines how the availability-sharing system fits into the interrupter's broader socio-technical context. Although a symmetrical system might help to mitigate power imbalances among collaborators by "implement[ing] some of the basic social rules that surround human face-to-face conversation" [43], asymmetrical systems allow colleagues to retrieve information about one another without requiring reciprocity in hardware deployment or necessitating explicit permission management. As a result, asymmetrical systems may lead to broader adoption and increased access to availability information within groups [45]. However, doing so without also implementing traceability may lead to negative social repercussions (e.g., a feeling that colleagues are "spying" on one another).

The degree of *obtrusiveness* is closely linked to the design decisions made regarding information delivery—"always on" systems generally require that additional hardware or computer display real estate be dedicated to the availability-sharing system. In some environments, this cost might be negligible, but especially for mobile individuals, it might be impractical to assume availability of a purpose-built display or peripheral installation. The optimal value for this axis might also be influenced by the context of the collaboration—tight coupling among colleagues might necessitate a more obtrusive design to foreground the activity of others. While both extremes might offer advantages, designing a system that can either provide output flexibility or a moderate degree of obtrusiveness is probably the best approach for supporting a broad range of information workers.

To assist an interrupter in determining when to initiate an interaction, a system can provide support at a variety of time scales—what we refer to as a *temporal gradient*. For example, an interrupter likely needs to know an interruptee's current availability information just before initiating an interaction. She may need to obtain additional context or a prediction of future availability if she discovers that it not a good time to begin a conversation. Although historical and recent information might provide anecdotal clues about a colleague's availability, the greatest benefit may result from being able to see current availability information or a prediction about the colleague's availability in the future.

Optimizing for the Interruptee

For the interruptee, the dimensions related to the way that the availability information is retrieved by the interrupter, including *information delivery* and *obtrusiveness*, are generally not relevant.

However, the *abstraction* of the interruptee's availability information collected by the system and the *presentation* approach used are perhaps the most important dimensions from the perspective of the interruptee. In general, sharing low-level media streams and sensor data, such as location, triggers a number of privacy concerns. More abstract availability measures have the benefit of mitigating some privacy concerns and reducing misinterpretations. For example, replacing "location" with "presence" or "proximity" to the office could reduce some of the privacy concerns of sharing location data. The presentation approach for this availability could be either abstract or discrete, as both approaches have proved useful in previous research.

Based on the principle of social translucence [7], we argue that *symmetry* is also of importance to the interruptee, since this property allows one to be aware of when and what information is shared and what values are shared with each collaborator. Because availability information may be privacy sensitive, contacts often regard each other at different levels of closeness (see Olson et al. on categories for sharing privacy sensitive information [37]). These varying closeness distinctions suggest that users might like to share availability information at different levels of granularity with different groups or individuals; they might even choose not to reciprocate sharing with a particular interrupter. While this consideration highlights the value of positioning the system to utilize asymmetric availability information exchange, a system that shares this information blindly creates the potential for spying and abuse of users' trust. The solution is to combine asymmetry with traceability, that is, keeping track of what information has been shared with others. These two features—categorized before as the asymmetric–traceable value in our design dimensions—aim at increasing confidence in the system, while also providing support for existing social practices.

Similarly, being able to examine an historical *temporal gradient* of the data captured, stored, and broadcast by the system can help the interruptee to build trust that the system is communicating accurate availability information. Prior awareness systems have placed significant emphasis on allowing users to exert control over the image that they project to others (e.g., [2]). In terms of the *temporal gradient* dimension, having reliable access to both a history of the data collected by availability-sharing systems and the current and predicted values that the system is broadcasting may be critical functionality for systems that rely on the placement of various kinds of sensing technologies in private or semi-private office spaces.

INTERRUPTME

In order to validate our design space as a useful tool for identifying under- or unexplored possibilities for supporting collaboration, we selected a unique combination of

Dimensions	Design Solution—InterruptMe			
Abstraction	Sensor	Avail.	Natural	MM
Presentation	Cont.	Discrete		Literal
Information Delivery	Always	Almost	Request	Implicit
Symmetry	S(T)	S(B)	A(T)	A(B)
Obtrusiveness	Focal	Selective	Appliance	Periph.
Temporal Gradient	Hist.	Recent	Current	Predict.

Figure 4: InterruptMe's design space classification.

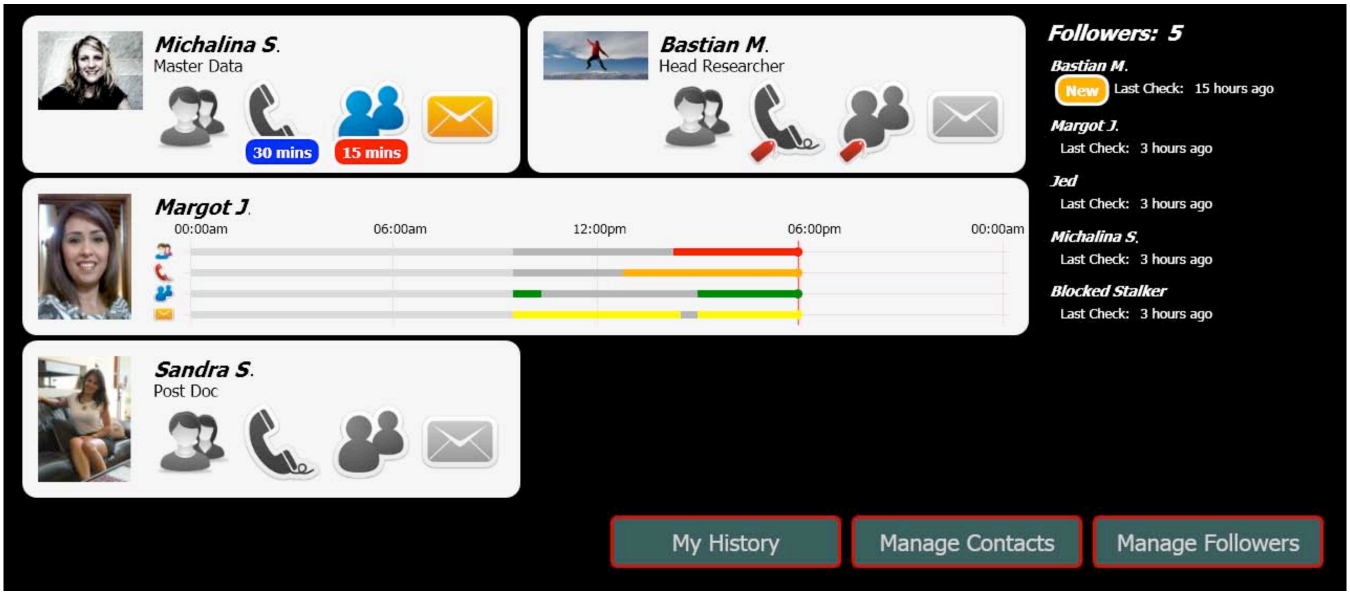

Figure 5: InterruptMe user interface. Color icons indicate availability and monochrome icons indicate unavailability. Tiles can be expanded to show availability history (center), and a "followers" list (right) promotes traceability for sharing.

points on each design axis to create a novel availability-sharing system, InterruptMe, guided by the design goals of balancing the needs of interrupters and interruptees (Figure 4). InterruptMe allows availability information to be shared on a per-colleague and per-communications medium basis, implements an asymmetric-traceable approach, and delivers information by means of implicit interaction.

On the interrupter's side, InterruptMe displays availability information about the interrupter's colleagues (Figure 5) using a projector-based display (Figure 6D). On the interruptee side, InterruptMe combines a suite of physical and electronic activity sensors (Figure 6B and C) to determine the interruptee's availability for collaboration over different kinds of communications media and shares this availability information according to the relationship to the interrupter. In this section, we present InterruptMe and its design rationale. Our design choices reflect a suite of hypotheses about the ways that availability-sharing systems are adopted and appropriated in the real world; we plan to evaluate them in depth in a future deployment study of the system.

InterruptMe addresses the *abstraction* dimension by using sensor data and computer activity information to model the availability of a user. The system differs from previous approaches in that it calculates a different availability value for each communications medium. This availability is shown according to a discrete *presentation*, i.e. the interruptee is either available or unavailable on each channel. The upper-left cell in Figure 5 shows the availability information of a user as four different values, one for each communication medium: the user is unavailable for face-to-face and phone interactions (monochrome icons with an optional blue history label), but she is available by instant messaging and email (full-color icons with optional red history label). This approach facilitates data interpretation by sharing an easily interpretable value for each communications medium, but does not disclose raw sensor or computer activity data, leaving little room for alternative or incorrect readings and preserving the interruptee's privacy.

InterruptMe uses a suite of sensors not only as information sources for our availability model, but also to unobtrusively bring the interface to the focus of the user, addressing the *information delivery* and *obtrusiveness* dimensions. InterruptMe avoids information overload and the effects of display blindness [20] by showing the interface only upon request (moving the mouse into the projector-based display) or when the user implicitly prepares to use a communications tool; the rest of the time, InterruptMe keeps the projection blank. The projected interface becomes visible when, for example, the user picks up their telephone hand-

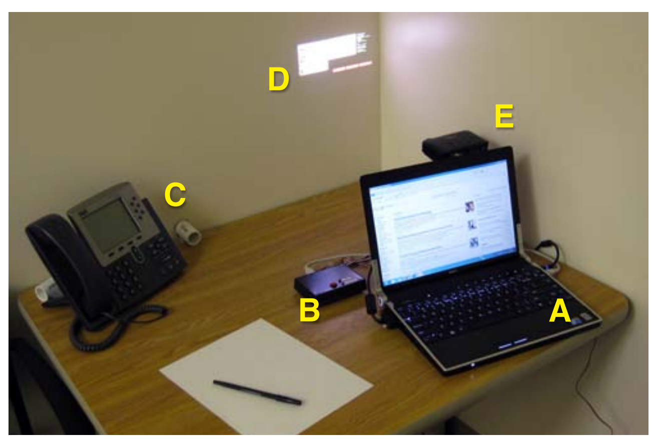

Figure 6: InterruptMe's physical setup. System components include (A) a computer, running the InterruptMe software and capturing electronic activity; (B) a Phidgets InterfaceKit module; (C) several motion, pressure, and contact sensors installed around the office space, and (D) the peripheral availability display, generated by (E) a tripod-mounted pico-projector.

set. InterruptMe detects that the interrupter is about to place a call and grays out those colleagues who are not currently available to receive phone calls. A similar response occurs when the user activates an IM application or email client on their desktop computer. By using implicit interaction, InterruptMe provides interrupters with availability information as an interaction is being initiated. This information helps them to decide whether to carry on with the intended interaction (when the user is available), to delay it, or to try a different communications medium. For example, if the interrupter lifts the telephone handset to call his colleague Michalina and sees the visualization shown in Figure 5 (in the upper-left cell), he might decide to IM or email her instead. However, our system is not proscriptive; the interrupter can always choose to go ahead and place the call.

InterruptMe addresses the symmetry dimension in two ways. First, in calculating the availability for each communications medium, InterruptMe takes into account the type of relationship existing between the users. InterruptMe builds upon Olson et al.'s categories of people for sharing sensitive information: *spouse*, *family*, *boss and trusted colleagues*, *other colleagues*, and *public* [37]. Each user classifies the people he or she shares information with according to these five groups (Figure 7A). The user can configure different levels of privacy for each communications medium and group (Figure 7B), choosing whether or not to share availability information and the level of detail at which to share. For example, the user in Figure 7B shares all availability information with the *spouse* group, and none with users in the *public* group; *family* (indicated by the green checkmarks), *boss and trusted colleagues*, and *colleagues* receive information only in relation to face-to-face and phone availability. When a group is configured not to receive information on a communications medium, the system shows that channel as being unavailable. This set of features position InterruptMe as an asymmetric system and aims at reducing some privacy-related problems by providing interruptees control over how and with whom their information is being shared.

The second way that the system addresses the symmetry dimension is by allowing non-reciprocal relations. User A can start following user B without B having to explicitly approve A's availability information request. In this case, user B will be notified about having a new *follower*. This new follower is placed, by default, in B's *public* group (see the public column of Figure 7A); B can later decide whether to move A to a more appropriate group, leave A in the public group, or block A from receiving further availability updates. B can also choose to reciprocate A's "friendship" and begin following A for access to his or her availability information. Moreover, the system keeps track of the last time A received information about B's availability. These features create a comprehensive traceable–asymmetry relationship between users. It is asymmetric because they do not have to assign the same privileges to one another, and it is traceable because each user is notified that they are being followed—and how often their information is requested.

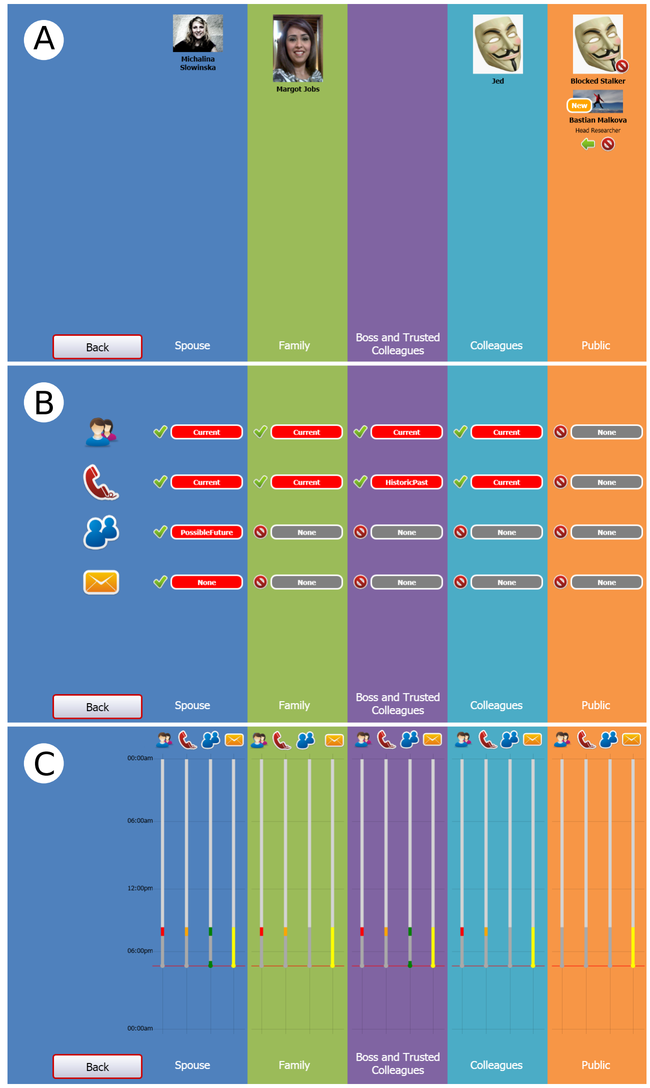

Figure 7: InterruptMe's privacy management interfaces: (A) assigning colleagues to groups, (B) choosing the availability information that each group can see; and (C) a summary of the resulting availability information shared during the current day.

Our system's asymmetry also supports the visibility characteristic of *socially translucent* systems, another of our design goals. To support mutual awareness of the information shared and to foster accountability, InterruptMe keeps track of one's own availability values throughout the day (Figure 7C) and of those received from other users. The third contact in Figure 5 has been expanded to reveal the changes of availability on each communications medium throughout the day. These historical values provide not only an increased level of social translucence, but also allow users to compare the information provided through the system with what they know about their contacts through other means, ideally leading to an increased degree of trust in the system.

InterruptMe is designed to require minimal active engagement from its users. In particular, users interact with the system only when they want to change their privacy settings, add contacts, or modify the relationships with their followers; InterruptMe provides an interface and sensible defaults for these operations (informed by previous systems and studies that we have described), allowing—but not requiring—users to fine-tune their settings.

Finally, building upon Greenberg's Peepholes system [14], InterruptMe allows users to actively monitor one another's availability, facilitating the "ambushing" behavior that is sometimes critical in dynamic workplace environments. Figure 5 shows small *tag* symbols that the local user has placed on the second contact's phone and IM channels. These tag icons indicate that when the contact becomes available through either of these communications media, a notification will be delivered to the local user.

InterruptMe differs from previous availability-sharing systems in three important ways. First, InterruptMe uses implicit interaction for displaying availability information, a characteristic previously seen only in systems that provide "walk-up" indications of availability. Second, InterruptMe implements traceable asymmetry, where interrupters can request availability information about their colleagues without having to be manually "approved" and where the interruptee does not have to symmetrically reciprocate in sharing availability information. Moreover, even though it is not explicit in the design space, InterruptMe introduces the notion of per-channel availability. These three differences set InterruptMe apart from prior availability-sharing systems and illustrate how our design space can be used to identify novel opportunities for designing availability-sharing systems and ameliorate previously identified problems (e.g., privacy, misinterpretation, and timely information delivery) in this class of systems.

CONCLUSION

In this paper, we explored the design space of availability-sharing systems. We used previous research on interruptibility and awareness systems to construct a new design space for availability-sharing systems. For each of the six axes in our design space—*abstraction*, *presentation*, *information delivery*, *symmetry*, *obtrusiveness*, and *temporal gradient*—we determined the range of values that each axis might take on, based on a survey of previous systems, and we discussed which values might be optimal for users with different roles in a collaboration.

We illustrated the utility of our design space in the design and implementation of InterruptMe, an opportunistic media-centered and person-dependent availability-sharing system. InterruptMe illustrates the utility of our design space: it introduces traceable asymmetry as a new point along a design space axis, and illustrates the use of pico-projector extended displays as a new way to realize an existing point in the space (information delivery by implicit interaction). We hope to complete a deployment study of this system in the future to validate its usefulness for supporting collaborations in the workplace.

InterruptMe is one illustration of a system designed specifically to balance between the needs of interrupters and interruptees, countering limitations of existing systems in this genre. The significance of our design space for availability-sharing systems is in enabling system designers to understand the important characteristics that define this class of systems and to think more concretely about how balancing these trade-offs in real-world collaborations.

ACKNOWLEDGEMENTS

This work was supported by the Danish Agency for Science, Technology, and Innovation under project #09-061856 the by the National Science Foundation under awards CNS-0937060 to the Computing Research Association for the Computing Innovation Fellows Project and IIS-0808783. We would also like to thank David Nguyen, Aurélien Tabard, Amy Voida, the STAR research group at UC Irvine, and our anonymous reviewers for their helpful feedback about this research.

REFERENCES

1. Bardram, J.E. and Hansen, T.R. The aware architecture: Supporting context-mediated social awareness in mobile cooperation. In *Proc. CSCW 2004*. ACM Press (2004), 192–201.
2. Begole, J.B., Matsakis, N.E., and Tang, J.C. Lilsys: Sensing unavailability. In *Proc. CSCW 2004*. ACM Press (2004), 511–514.
3. Biehl, J.T., Turner, T., Quarfordt, P., van Melle, B., Dunnigan, T., and Golovchinsky, G. MyUnity: Building awareness and fostering community in the workplace. http://adsabs.harvard.edu/abs/2010arXiv1006.5024B.
4. Cadiz, J.J., Venolia, G., Jancke, G., and Gupta, A. Designing and deploying an information awareness interface. In *Proc. CSCW 2002*. ACM Press (2002), 314–323.
5. Dabbish, L. and Kraut, R.E. Controlling interruptions: Awareness displays and social motivation for coordination. In *Proc. CSCW 2004*. ACM Press (2004), 182–191.
6. Dourish, P. and Bly, S. Portholes: Supporting awareness in a distributed work group. In *Proc. CHI 1992*. ACM Press (1992), 541–547.
7. Erickson, T. and Kellogg, W.A. Social translucence: An approach to designing systems that support social processes. *ACM Trans. Computer–Human Interaction, 7,* 1 (2000), 59–83.
8. Fetter, M., Seifert, J., and Gross, T. Lightweight selective availability in instant messaging. In *Ext. Abstracts CHI 2010*. ACM Press (2010), 3817–3822.
9. Fish, R.S., Kraut, R.E., Root, R.W., and Rice, R.E. Evaluating video as a technology for informal communication. In *Proc. CHI 1992*. ACM Press (1992), 37–48.
10. Fitzpatrick, G. *The Locales Framework: Understanding and Designing for Wicked Problems*. Kluwer, Dordrecht, The Netherlands, 2003.
11. Fogarty, J., Hudson, S.E., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C., and Yang, J. Predicting human interruptibility with sensors. *ACM Trans. Computer–Human Interaction, 12,* 1 (2005), 119–146.
12. Fogarty, J., Lai, J., and Christensen, J. Presence versus availability: The design and evaluation of a context-aware communication client. *International Journal of Human–Computer Studies, 61,* 3 (2004), 299–317.

13. Garcia Vazquez, J.P., Rodriguez, M.D., and Andrade, A.G. Design dimensions of ambient information systems to assist elderly with their activities of daily living. In *Ext. Abstracts UbiComp 2010*. ACM Press (2010), 461–464.

14. Greenberg, S. Peepholes: Low cost awareness of one's community. In *Ext. Abstracts CHI 1996*. ACM Press (1996), 206–207.

15. Gross, T. and Oemig, C. From PRIMI to PRIMIFaces: Technical concepts for selective information disclosure. In *Proc. SEAA 2006*. IEEE Computer Society (2006), 480–487.

16. Harr, R. and Wiberg, M. Lost in translation: Investigating the ambiguity of availability cues in an online media space. *Behaviour & Information Technology, 27,* 3 (2008), 243–262.

17. Horvitz, E. and Apacible, J. Learning and reasoning about interruption. In *Proc. ICMI 2003*. ACM Press (2003), 20–27.

18. Hsieh, G., Tang, K.P., Low, W.Y., and Hong, J.I. Field deployment of IMBuddy: A study of privacy control and feedback mechanisms for contextual IM. In *Proc. UbiComp 2007*. Springer (2007), 91–108.

19. Hudson, J.M., Christensen, J., Kellogg, W.A., and Erickson, T. "I'd be overwhelmed, but it's just one more thing to do": Availability and interruption in research management. In *Proc. CHI 2002*. ACM Press (2002), 97–104.

20. Intille, S. Change blind information display for ubiquitous computing environments. In *Proc. UbiComp 2002*. Springer (2002), 193–222.

21. Isaacs, E., Walendowski, A., and Ranganthan, D. Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In *Proc. CHI 2002*. ACM Press (2002), 179–186.

22. Isaacs, E.A., Tang, J.C., and Morris, T. Piazza: A desktop environment supporting impromptu and planned interactions. In *Proc. CSCW 1996*. ACM Press (1996), 315–324.

23. Jonsson, M., Jansson, C., Lönnqvist, P., Werle, P., and Kilander, F. Achieving non-intrusive environments for local collaboration. Technical Report 2002-021, Department of Computer and Systems Sciences, Stockholm University/KTH, Stockholm, Sweden (February 2008).

24. Lai, J., Levas, A., Chou, P., Pinhanez, C., and Viveros, M. BlueSpace: Personalizing workspace through awareness and adaptability. *International Journal of Human–Computer Studies, 57,* 5 (2002), 415–428.

25. Lai, J., Yoshihama, S., Bridgman, T., Podlaseck, M., Chou, P., and Wong, D. MyTeam: Availability awareness through the use of sensor data. In *Proc. INTERACT 2003*. IOS Press (2003), 503–510.

26. MacIntyre, B., Mynatt, E.D., Voida, S., Hansen, K.M., Tullio, J., and Corso, G.M. Support for multitasking and background awareness using interactive peripheral displays. In *Proc. UIST 2001*. ACM Press (2001), 41–50.

27. Mark, G., Gonzalez, V.M., and Harris, J. No task left behind?: Examining the nature of fragmented work. In *Proc. CHI 2005*. ACM Press (2005), 321–330.

28. Mark, G., Gudith, D., and Klocke, U. The cost of interrupted work: More speed and stress. In *Proc. CHI 2008*. ACM Press (2008), 107–110.

29. Markopoulos, P. A design framework for awareness systems. In P. Markopoulos, B. de Ruyter, and W. Mackay (Eds.), *Awareness systems: Advances in theory, methodology and design*. Springer, Dordrecht, The Netherlands (2009), 49–72.

30. Markopoulos, P., de Ruyter, B., and Mackay, W. *Awareness systems: Advances in theory, methodology and design*. Springer, Dordrecht, The Netherlands (2009).

31. Matthews, T., Dey, A.K., Mankoff, J., Carter, S., and Rattenbury, T. A toolkit for managing user attention in peripheral displays. In *Proc. UIST 2004*. ACM Press (2004), 247–256.

32. McEwan, G. and Greenberg, S. Supporting social worlds with the community bar. In *Proc. GROUP 2005*. ACM Press (2005), 21–30.

33. McFarlane, D.C. Interruption of people in human-computer interaction: A general unifying definition of human interruption and taxonomy. Technical Report NRL/FR/5510-97-9870, Naval Research Laboratory, Washington, DC (1997).

34. McFarlane, D.C. and Latorella, K.A. The scope and importance of human interruption in human-computer interaction design. *Human–Computer Interaction, 17,* 1 (2002), 1–61.

35. Milewski, A.E. and Smith, T.M. Providing presence cues to telephone users. In *Proc. CSCW 2000*. ACM Press (2000), 89–96.

36. Nichols, J., Wobbrock, J.O., Gergle, D., and Forlizzi, J. Mediator and medium: Doors as interruption gateways and aesthetic displays. In *Proc. DIS 2002*. ACM Press (2002), 379–386.

37. Olson, J.S., Grudin, J., and Horvitz, E. A study of preferences for sharing and privacy. In *Ext. Abstracts CHI 2005*. ACM Press (2005), 1985–1988.

38. Patil, S. and Kobsa, A. Instant messaging and privacy. In *Proc. HCI 2004*. British Computer Society (2004), 85–88.

39. Pousman, Z. and Stasko, J. A taxonomy of ambient information systems: Four patterns of design. In *Proc. AVI 2006*. ACM Press (2006), 67–74.

40. Tang, J.C., Isaacs, E.A., and Rua, M. Supporting distributed groups with a montage of lightweight interactions. In *Proc. CSCW 1994*. ACM Press (1994), 23–34.

41. Tang, J.C., Yankelovich, N., Begole, J., van Kleek, M., Li, F., and Bhalodia, J. ConNexus to Awarenex: Extending awareness to mobile users. In *Proc. CHI 2001*. ACM Press (2001), 221–228.

42. Tomitsch, M., Kappel, K., Lehner, A., and Grechenig, T. Towards a taxonomy for ambient information systems. Position paper presented at the Workshop on the Issues of Designing and Evaluating Ambient Information Systems at PERVASIVE 2007, Toronto, ON (2007).

43. Vertegaal, R., Dickie, C., Sohn, C., and Flickner, M. Designing attentive cell phone using wearable eyecontact sensors. In *Ext. Abstracts CHI 2002*. ACM Press (2002), 646–647.

44. Voida, A., Newstetter, W.C., and Mynatt, E.D. When conventions collide: The tensions of instant messaging attributed. In *Proc. CHI 2002*. ACM Press (2002), 187–194.

45. Voida, A., Voida, S., Greenberg, S., and He, H.A. Asymmetry in media spaces. In *Proc. CSCW 2008*. ACM Press (2008), 313–322.