Context–Aware Composition

Kent Lyons, Roy Want, David Munday[†], Jiasheng He[‡],
Shivani Sud, Barbara Rosario, Trevor Pering

Intel Research
2200 Mission College Blvd.
Santa Clara, CA 95054
{kent.lyons, roy.want, shivani.a.sud, barbara.rosario, trevor.pering}@intel.com

[†]Baskin School of Engineering
UC Santa Cruz
Santa Cruz, CA 95064
cromom@soe.ucsc.edu

[‡]School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332
jiazhe@cc.gatech.edu

ABSTRACT

Context–Aware Composition is a technique to overcome the complexity of wireless discovery, device selection, and establishing connections to other computers in an environment where potentially tens or even hundreds of other computers may reside. In this paper, we present our work demonstrating the benefit of using context to facilitate the discovery process through sensing. We describe our sensor platform, the Context Card, and discuss how the Sensor Actuator API (SAAPI) is used to collect and distribute sensor data. We also show how context can be used to create multi-modal composition commands and provide a quantitative evaluation of a multimodal position sensing system to support context-aware composition.

INTRODUCTION

A major trend that defines the modern digital era is the proliferation of a wide variety of heterogeneous mobile computers carried by users, co-existing with both a wired and wireless computing infrastructure. The proliferation of WiFi based PCs, netbooks, smart phones, CE devices, printers, scanners, Bluetooth peripherals, and GPS units is continuing unabated, and is giving rise to a new kind of wireless problem: making sense of what is discovered wirelessly from a long list of device names, typically without any rational way for a user to distinguish one from another. Today there are no standard mechanisms for providing user friendly names for devices or network identifiers. Furthermore, some users do not change the name of their devices, resulting in many devices which simply use the default manufacturer's name and are textually indistinguishable, only separable by their obscure MAC addresses.

To make matters worse, each radio standard, such as WiFi and Bluetooth or the emerging Ultra-WideBand (UWB) and WiMax standards, all are based on very different physical radios, and typically have independent connection management programs each with its own connection models and idiosyncrasies. Without a coordinated approach, this proliferation of radio technologies turns a wealth of communication opportunities into a confusing maze of obscure errors and connection troubleshooting. In most situations today, the problem is typically still manageable, but looking to the future, it will become a more significant issue as a wider variety of wireless devices are introduced and additional wireless technologies are embedded in everyday items.

Such issues of scale will be paramount for any application that seeks to rapidly connect devices together in an environment where many other wireless devices exist. Techniques such as Dynamic Composable Computing (DCC) [13] and Pervasive Collaboration [9] seek to enable compositions (or collections) of mobile components in order to combine them together to form aggregate computing platforms that are more capable of carrying out a user task. However, in environments such as a crowded cafe or office, there could easily be tens or even hundreds of other electronic devices nearby, providing a bewildering array of target devices for users to select from when forming a composition. Systems operating in such environments all face the same challenge of scale seen during conventional wireless discovery, and similarly needs a suitable solution: the problem is not necessarily the limited number of devices that the users wish to combine together, rather, it is the numerous other nearby devices that are not intended to be part of the collection.

In this paper we explore how context, specifically information obtained through a mobile device augmented with numerous sensors, can facilitate the discovery process. Such sensors can provide informative capabilities for mobile devices ranging from novel gesture-based interaction techniques to activity recognition. However, in this work, we explore how sensors can also be used to facilitate discovery through spatial sensing, and by representing unique aspects of their state, such as the direction they are facing or if they are in motion. This additional information can provide users with distinguishing characteristics that allow them to better distinguish each device. Further, it provides additional

HotMobile 2009, February 23–24, 2009, Santa Cruz, CA, USA. Copyright 2009 ACM 978-1-60558-283-2/09/02...$5.00.

parameters to support filtering which can then be used to limit the selection of devices that are presented to a user.

The research contributions of this work are: first, the novel use of sensors to facilitate discovery using a graphical representation of their state and providing information that can be used to filter input to a Dynamic Composable Computing system; second, a generalized sensor-bus architecture that can be integrated with a variety of wireless discovery mechanisms, and simultaneously supporting context-aware applications that wish to have access to sensor data. We discuss the use of sensors in both connection-less (discovery) and connected (IP data-link) modes of use between mobile devices, and present a quantitative evaluation of a positioning system that supports context-aware multi-display composition; a mechanism for wirelessly combining the screen real-estate of several computers into a single coordinated display surface.

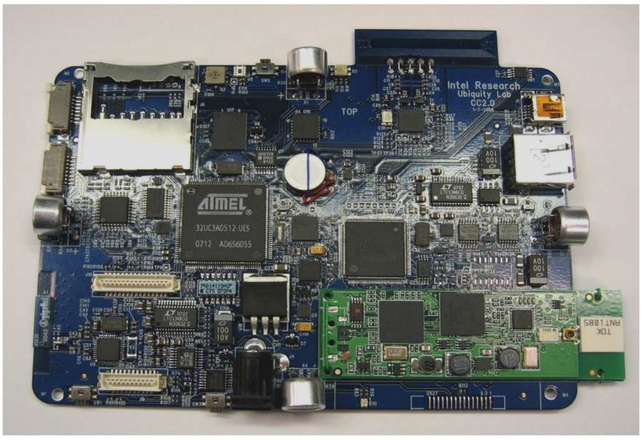

Figure 1. The Context Card with 4x ultrasound, 4x IrDa, magnetometer, 2D Gyro, 2x 3D accelerometers, light sensor, barometer, capacitive touch, IEEE 802.15.4, NFC, UWB and Bluetooth.

ARCHITECTURE FOR CONTEXT-AWARE COMPOSITION

To explore the issues related to context-aware composition, we created a system composed of several components: the Context Card, the Sensor Actuator API (SAAPI), and the Composition Framework. The Context Card is a custom circuit board supporting numerous sensors designed to attach to the back of a UMPC. SAAPI is a pair of programs, one running on the context card and the other on the attached host computer, that together provide a mechanism for controlling and acquiring sensor data. Finally, the Composition Framework provides the mechanisms needed to create a logical computing platform from multiple platform services.

Context Card

The context card is an embedded sensor platform (Figure 1) designed to attach to a mobile computer, and runs the embedded SAAPI interpreter. Targeting the Sony Vaio UX90 UMPC, we designed the context card to have the same physical dimensions as the mobile device. This size allows us to mount the card directly to the back of the UMPC using its existing mounting holes. It also provides real estate for many different kinds of sensors that acquire data about device motion, the spatial layout of devices, and environmental parameters. The context card also has several radio interfaces including Near Field Communication (NFC), ultra-wideband (UWB), Bluetooth, and either an IEEE 802.15.4 or Zigbee radio that allows us to explore power-bandwidth trade-offs of multi-radio systems [8].

The context card contains two 3-axis accelerometers and a 2-axis gyroscope which sense linear acceleration and rotational velocity respectively. It also has a magnetometer to sense its orientation with respect to the Earth's magnetic field. Together these sensors provide information about the physical movement and orientation of the attached mobile computer. It has an ultrasonic transducer on each of its four edges. This configuration enables the device to emit and receive ultrasonic pulses from any direction, or combination of directions, and to determine the bearing of incoming pulses. By measuring the time–of–flight of pulses sent between devices, and knowing the speed of sound in air at room temperature, their spatial separation can be calculated. Furthermore, with several devices participating, tri–lateralization can be employed to obtain the spatial topology [2]. In addition to ultrasonic ranging, the context card has an IrDA transceiver at the center of each of its four edges. While IrDA does not provide ranging, this configuration provides for out-of-band discovery/identity (independent of a radio), and coarse grained information about relative device directionality (e.g. up, down, left, right), as does the ultrasound system. Finally, the context card has several sensors to measure ambient environmental conditions including a barometer, a thermometer, and a visible/infrared light sensor.

Sensor Actuator API (SAAPI)

The goal of the SAAPI design is to allow applications to concurrently learn about the sensors and actuators that are available on a platform, register interest in particular sensor types, and receive streams of data or query/command/responses reflecting their state. Further, a system goal is to represent sensor data as textual key/value pairs that can be parsed by a variety of higher-level programs using a generic parsing process. Therefore, the SAAPI abstraction is responsible for 1) hiding the wide variety of wire-line protocols used by the various types of sensors, for example I2C and SPI, and the specific set up and reporting protocol established over these buses; 2) enabling many different sensor types, or instances of sensors to co-exist on the same communication bus; 3) offloading the real-time sensor interaction task from the host processor, and only reporting information that is time or status critical (e.g. responding to an event or crossing a predefined threshold). The latter capability enables improved power consumption at the host, which can remain in a low-power sleep-mode for a longer period of time.

On the host device, SAAPI connects to the context card's serial data port to send and receive control requests and sensor data. The host also provides an optional interpreter

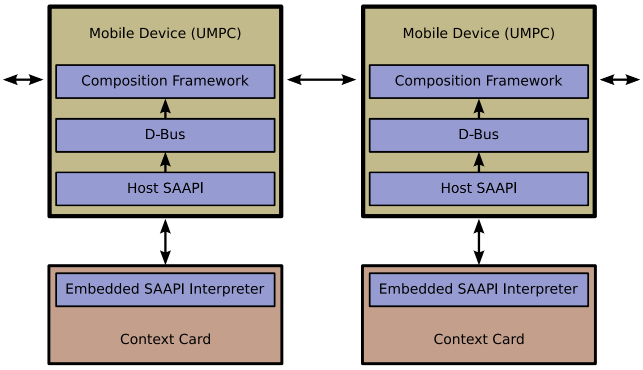

Figure 2. Components of Context Aware Composition System.

for interactive debugging. SAAPI acts as a bridge between the sensor data obtained from the context card and the rest of the system. In particular it uses D–Bus[1] as an inter-process communication mechanism. D–Bus is commonly used by many Linux systems and provides a publish/subscribe model of event notification for information sharing among the various event generators (sensors) and consumers (applications like the Composition Framework, described in the next section). SAAPI registers a pair of signals on D–Bus so that any other local application can initiate sensing requests, or receive any collected data. The embedded interpreter on the context card processes string-based command requests for sensor data. SAAPI implements synchronous commands which can be used to request a single reading from a given sensor. SAAPI also has asynchronous capabilities which are used when a sensor event is generated such as when one of the push-buttons changes state. The asynchronous capability is also used for streaming sensor data. In this mode, the embedded software regularly samples sensor data and reports it to the host.

Composition Framework

The Composition Framework is the core of our composition system [13] which is responsible for performing device and platform service discovery. Furthermore, it provides a control mechanism for instantiating the underlying client and server software needed to create compositions. The Composition Manager (Figure 3) is a graphical user interface that shows a global view of discovered devices and services, depicts instantiated compositions, and provides mechanisms for a user to alter the state of the composed systems. The Composition Framework can either utilize traditional multi-cast discovery mechanisms for IP based networks, or extend the system to support Link Layer (L2) discovery of devices and services. It also serves as a communication hub managing data across a device's numerous network interfaces (WiFi, UWB, Bluetooth, etc.). Finally, the Composition Framework connects to the SAAPI D–Bus interface as a subscriber to receive information generated by the sensors which is interpreted and shown as needed as part of the user interface (Figure 2). While other sensing sub-systems such as SeeMon [6] have been described in the literature, they have not been designed to support wireless discovery or to conveniently integrate with an existing and well established IPC mechanisms such as D-Bus which may facilitate wide adoption.

[1] http://www.freedesktop.org/wiki/Software/dbus

CONNECTION-LESS CONTEXT-BASED DISCOVERY

This combination of the context card, SAAPI, and the Composition Framework enables many new interaction capabilities. In particular, by using layer-2 to distribute a device name as well as service and context information, the combined discovery and connection time can be greatly optimized [11]. Specifically, service information can be embedded in the Application Specific Information Elements (ASIEs) of the beacons employed by either UWB radios or 802.11n. With layer-2 advertisement, we publish relevant context data over the link layer of the radio without requiring a MAC layer association among the peers. This use of layer-2 greatly reduces the overhead imposed by traditional methods that share information over layer 3. Our implementation uses this capability to distribute relevant information about composition–enabled services to peer devices. With the inclusion of sensing, interpreting, and distributing of context information, these layer-2 service advertisements become a source of more semantically relevant information about available devices and services.

Discovered Device Representations

There are numerous types of user interface representations that can be used to show wirelessly discovered devices. One very common technique is to provide a simple list of discovered devices. For example, a Bluetooth or WiFi scan might return a list of "nearby" device names. Unfortunately, those names may or may not contain useful information for identifying a device of interest [12]. Furthermore, this technique does not scale well as the number of devices increases. Some devices are also capable of providing a limited notion of distance, for example using signal strength, which then allows the list of discovered resources to be sorted (approximately) by distance from the user.

Distance can also be explicitly measured using a variety of techniques. Ultrasonic ranging, for example, can be used to determine the distance between devices equipped with transducers, and with a sufficient number of devices, tri–lateralization can be used to determine the spatial topology [2]. This spatial information can be presented in a user interface creating a virtual representation of the devices in the physical space around a user [7]. Research simulating this type of spatial sensing has shown that for a small number of devices the spatial (iconic) representations tend to offer better user performance for identifying devices [1].

In addition to the above spatial sensing capabilities, we are exploring how additional sensors might further improve the ability for a user to make the mental connections between physical devices of interest and their associated, likely ambiguous, discovered representations. Our strategy is to use our connection-less context sharing capabilities to distribute semantically useful information for presentation in a user interface along with the automatically discovered devices. For example, consider a scenario in which a user is in a rich digital environment where dozens of devices have been discovered but only one specific device is of interest.

By using our layer-2 context sharing, we can provide extra information to help overcome this problem. For example, the user can pick up the mobile device of interest and the system can graphically highlight the device that reported movement. The Context Framework uses information from the attached accelerometers to determine movement and broadcasts a context event to other nearby devices. The Context Framework on the receiving end receives this event, and in turn "shakes" the graphical representation on the screen. By watching her display, the user can now see which of the discovered devices corresponds to the physical device of interest. This concept is similar to other work using shared context to connect two devices [3]; however, we are using it to inform the user and represent the information in the discovery process. Other sensors can also provide cues to the user; Figure 3 depicts the use of a magnetometer where the orientation of a device (shown by the red triangle) is used for identification.

In addition to graphically depicting context data, the information can be used to filter irrelevant devices. Spatial sensing can be used to only show nearby devices, the motion sensors might be used to filter out any non-static devices, etc. The data from one sensor can be used individually, or multiple sensing modes can be used in conjunction. Likewise, multi-modal interfaces can provide additional mechanisms for representing and controlling this filtering process. We can use a mobile's touchscreen to express vicinity (by drawing a circle around the device of interest for example) or a voice command by saying "show me nearby devices" or "show me the fixed infrastructural devices." Each of these pieces of information provide additional semantically relevant information to a user and can be used individually or in concert to narrow down the range of discovered devices.

Multi-Modal Composition Commands

Our implementation of the context–aware composition system also enables sensor-based interaction techniques for manipulating compositions. For example, inspired by a "point and shoot" metaphor, we created a simple gesture-based interaction that allows a user to rotate her mobile device, and with a quick shake downward, create a connection between her device and another composition-enabled computer.

For this interaction, we use the magnetometer to sense the orientation the mobile device. As the user turns the device, the graphical representation of its orientation also changes among the discovered devices. When it is pointing towards the computer with the service of interest, the user performs the second part of the interaction, a downward shake. This gesture is sensed using data from the z-axis of one of the accelerometers. If the system detects the shake event, and the orientation is such that the device is pointing towards another (and not just empty space), a connection is created (yellow link, Figure 3). To disconnect the composition, we employed a second metaphor and gesture based command.

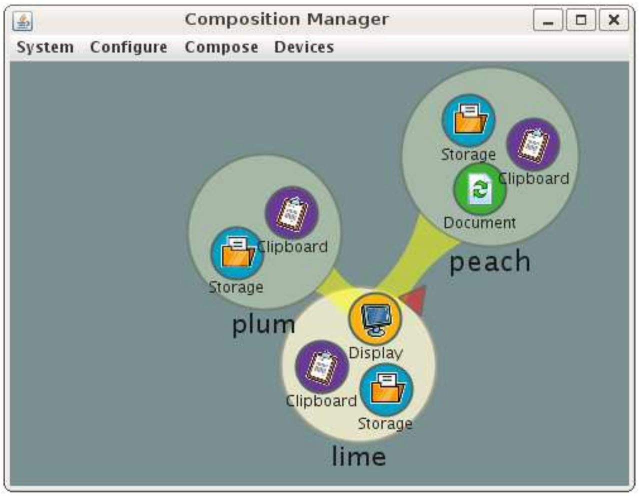

Figure 3. The red triangle depicts orientation information for the device called "lime" sensed by the magnetometer. Similarly, the graphical representation of a device shakes (moving side to side) when it is picked up as sensed by the accelerometers.

By inverting the mobile device, and shaking it side-to-side, the user performs a gesture similar to the movement needed to erase the "Etch-a-sketch" (a popular sketching toy for children).

We have also designed and implemented a speech interface for creating compositions using (purely) voice commands. A logical extension to this capability is to leverage the sensor data to enable additional multi-modal interactions. For example, a user might select a device using the sensors (by conveniently orienting them as described above, for example) and use a voice command to initiate the connection. Voice has the further advantage of allowing the user to provide additional semantic information to the system and allow for the specification of the desired service.

There are two important aspects of these types of command gestures. First, the architecture and implementation of the Composition Framework enables these commands to be sensed and used to initiate IP connections to other devices. Second, we use multiple dimensions of sensed data and specifically designed gestures to help with the segmentation and recognition process.

CONTEXT ENABLED COMPOSITION SERVICES

In addition to using context as a mechanism for the user to better manage the discovery process and for controlling compositions, context can also serve a useful role for the services involved in a composition. For example, information about the relative spatial position of devices would be useful for a keyboard and mouse sharing service. Projects such as the iRoom [4] included the ability for a single mouse (and keyboard) to logically move from one device to another; however, the mapping between devices was performed manually [5]. By utilizing our composition engine, a similar configuration of devices could be assembled in an *ad hoc* fashion. Furthermore, the inclusion of sensing could

Figure 4. Four separate tablet computers, configured as a multi-display composition by the Composition Framework and linked by a wireless network, appears as a single logical display.

allow for the automatic determination of the logical spatial connections between devices.

In our own work, we are exploring how a display composition can be formed out of several mobile touch–screen devices. Instead of just mirroring the display from one device to another as achieved with a single client and server, we have created a multi-client composition [13] which utilizes a display sharing service to create a single logical display spread across several mobile devices (Figure 4). Each client in the composition shows a different region of the overall display. While there has been work on statically tiling displays using a wired network or developing codecs for video playback across multiple mobile devices [10], our system, coordinated by the Composition Framework, is designed as a general purpose *ad hoc* desktop sharing solution which can extend the display of several mobile devices connected over a wireless network to form a single logical display in any topology.

Our current implementation of the multi-client display service logically attaches the displays in raster scan order (left to right, top to bottom). While this creates a functional system, the user must be careful about the order in which the displays are connected and their physical location relative to each other. If the displays are connected in the wrong order or moved, the logical display is no longer contiguous even if the devices are still in physical proximity to each other. However, the underlying software framework is flexible enough to allow them to compose in any topology defined by a preferred policy, which includes mimicking the relative positioning of the physical displays.

Context-Aware Multi-Display Composition Evaluation

We have begun the integration of context data into our multi-display composition system. Here, we present results from experiments designed to gain insight into how the ultrasonic ranging capabilities of the context cards could be used for updating the relative spatial positions of the displays. Our first set of evaluations is designed to characterize the ability for multiple context cards to perform the measurements needed for our tiled display composition. In particular, we are interested in the ability of the system to report relative device orientation (e.g. the Left edge of device 1 is facing the Bottom edge of device 2). Furthermore, we are interested in characterizing the ability of the devices to perform ranging measurements between each other.

For these tests, two context cards are attached to one computer. That computer runs software which connects to the SAAPI interface of each context card over D-Bus. Further, it issues commands to perform the orientation and ranging between the two context cards, and it logs results from the test.

The software performs a range operation between the context cards approximately every 150ms. For a given range measurement, one context card serves as an active repeater while the other initiates the ranging operation and measures the time between when the initial ultrasonic ping was sent and the time an active echo was received. Using this time, the context card reports a computed distance. Furthermore, both context cards report which of the four ultrasonic transducers (top, bottom, left, right) first received the signal. This process continues with the software swapping device rolls at each iteration.

Our first test focuses on the accuracy of directionality determination and stability of ranging measurements. For this test, two devices were placed approximately 15cm from each other and remained static. The test lasted approximately 30s. During that time, the system performed 175 ranging measurements, and all 175 results returned the correct orientation of the cards relative to each other. The mean sample rate was 152.1ms (SD=12.7). The average time to complete a single measurement was 95.4ms (SD=8.7). The mean distance computed was 16.7cm (SD=0.36). There was also no statistically significant difference between the directionality of the ranging (M=16.7cm SD=0.46cm initiating from card one, M=16.7cm, SD=0.21 from card two) as computed using a two-tailed Students T-test (t=0.46, p=0.65).

For our second test, we wanted to measure how quickly the system could determine the relative orientations of the cards once a user picked one up and placed it in a new position. For this test, one card remained static while the other card was moved by a researcher clockwise, starting at the top position. Once data collection started, the card was moved to the left of the static context card and remained there for approximately 5 seconds. The researcher then moved the card to the bottom and let it remain there for approximately 5 seconds. This process continued until the context card made the final transition and was placed again in the starting position. To determine when the card was stationary and when it was moved, we used the on-board accelerometers and reported movement events in our log files along with the ranging and orientation measurements. We post-processed the logs and determined that movement events less than one second apart were part of the same motion, while the remaining time represents the static portion of the experiment. For this experiment we are interested in the accuracy of determining orientation and how quickly the system could detect it.

We conducted two trials of this test. For both trials, the system correctly determined the relative orientation of the two devices during the static portion of the experiment with 100% accuracy (87 of 87 readings for trial one and 67 of 67 readings for trial two). Furthermore, by the time the device was finished being moved, the system had already determined the new orientation. On average this determination was made 1.6s (SD=0.56) before the card repositioning was completed by the researcher.

These results indicate that the system is capable of very accurately determining the relative orientation and distance between two devices. Furthermore, the system quickly adapts to the movement of the devices and can correctly determine the new relative orientation even before the move operation is complete. These are important first steps for being able to automatically determine the relative tiling of the devices forming a multi-display composition.

FUTURE WORK AND CONCLUSIONS

To date we have created many of the system components described here, and have begun linking them to support generalized context-aware composition capabilities. At this time sensor data from the compass and accelerometer is passed from the Context Card through the SAAPI D–Bus interface and Composition Framework, where it is displayed in a user interface. This allows the discovery process to be combined with sensor data and represented graphically.

The point-and-shoot prototype, including the gestural rules for connection, has been created as an application on top of this system. However, creating a single integrated system that generalizes all of these features is the subject of future work. Nevertheless, the SAAPI interface and the D-Bus inter-process communication mechanism have proven suitable for augmenting discovery, while at the same time providing a mechanism for context-aware applications to tap into sensor data of interest. Future work includes the extension of this mechanism to support multi-radio interfaces, integrating discovery and sensor data into a single user interface that can also serve to initiate connections and build compositions across heterogeneous wireless interfaces. Given our success with the discovery point-and-shoot application, it warrants an exhaustive study of other common sensors, typified by those on the context card, that can also provide multi-modal support for composable systems.

The data and results generated by our evaluation of the context-aware multi-display composition are also promising. Our test setup provides indications that the context cards will be able to perform the orientation measurements needed to determine the physical tiling of displays. Future work will involve incorporating the ranging information into the Composition Manager and multi-display server, testing the scalability of the system with more devices and improving the underlying ranging capabilities.

Our initial work in this area indicates that context-aware discovery has significant potential for overcoming many of the issues we will soon encounter as numerous wireless devices and services become available in everyday situations.

REFERENCES

1. R. Gostner, E. Rukzio, and H. Gellersen. Usage of spatial information for selection of co-located devices. In *Proc. of MobileHCI '08*, pages 427–430, 2008.
2. M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. In *Proceedings of MobiSys '05*, pages 177–190, 2005.
3. L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Proceedings of UbiComp '01*, pages 116–122, 2001.
4. B. Johanson, A. Fox, and T. Winograd. The interactive workspaces project: Experiences with ubiquitous comput ing rooms. *IEEE Pervasive Computing*, 1(2):67–74, 2002.
5. B. Johanson, G. Hutchins, T. Winograd, and M. Stone. Pointright: experience with flexible input redirection in interactive workspaces. In *Proceedings of UIST '02*, pages 227–234, 2002.
6. S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song. Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proceeding of MobiSys '08*, pages 267–280, 2008.
7. G. Kortuem, C. Kray, and H. Gellersen. Sensing and visualizing spatial relations of mobile devices. In *Proceedings of UIST '05*, pages 93–102, 2005.
8. T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of MobiSys '06*, pages 220–232, 2006.
9. T. Pering, R. Want, B. Rosario, S. Sud, and K. Lyons. Enabling pervasive collaboration with platform composition. In *Proceedings of Perviasive '09*, 2009.
10. G. Shen, Y. Li, and Y. Zhang. Mobius: enable together-viewing video experience across two mobile dev ices. In *Proc. of MobiSys '07*, pages 30–42, 2007.
11. S. Sud, R. Want, T. Pering, B. Rosario, and K. Lyons. Rapid wireless service composition using layer-2 discovery. *IEEE Network Magazine*, July–Aug 2008.
12. T. J. Tim Kindberg. "merolyn the phone": A study of bluetooth naming practices. In *Proceedings of Ubicomp*, pages 318–335, 2007.
13. R. Want, T. Pering, S. Sud, and B. Rosario. Dynamic composable computing. In *Proceedings of HotMobile 2008*, February 2008.