Kimono: Kiosk-Mobile Phone Knowledge Sharing System

Albert Huang[1] Kari Pulli[1,2] Larry Rudolph[1]

[1]Massachusetts Institute of Technology [2]Nokia Research Center

Abstract

The functionality of an information kiosk can be extended by allowing it to interact with a smartphone, as demonstrated by the Kimono system, and the user interface can be greatly simplified by "associations" between pieces of information. A kiosk provides information that is relevant to a particular location and can use valuable context information, such as the fact that a user is physically standing in front of the kiosk, to tailor the display. Its graphically rich screen is suitable for presenting information to the user and has a natural input modality requiring the user to simply touch the screen. However, a kiosk lacks mobility and cannot stay with the user as he or she moves about the environment. Also, information provided by the kiosk must be remembered by the user. Finally, it is difficult to add information to the kiosk, and so the kiosk remains an information display device.

All this changes when a handset, such as a PDA or smartphone, can interact with the kiosk. The handset acts like a personalized proxy of the kiosk. It accompanies the user serving as a memory device. It is also an excellent media creation device, capable of taking pictures and recording voice memos as well as short text messages. Associating newly created content with other currently selected content makes for a simpler user interface. Content and its associations can be uploaded to a kiosk allowing others to access to it.

1 Introduction

This paper presents Kimono, **Ki**osk-**mo**bile phone k**no**wledge sharing system. Kimono is about the organization, display, and distribution of information that is most pertinent to the current time and place of the user. The system is adapted for two different modes of access—that of a kiosk computer with a generously sized display and a robust interface, and that of a personal mobile device that typically accompanies the user, such as a PDA or mobile phone. The acquisition and exchange of data and information is designed to be as simple and efficient as possible, prompting the user for decisions only when absolutely necessary, and exchanging only information that is determined to be relevant to the user.

The field of ubiquitous computing [Weiser 1991] studies how computers move from desktops, places where people go to work and access computing, to places where people live

Figure 1: The OK-net terminal with a touch panel display and an optional keyboard.

and interact with other people. The Oxygen Kiosk Network (OK-net) [OKn ; Rudolph 2001; Van Kleek 2003a] is an effort to bring the promises of ubiquitous computing to public and transient spaces in the workplace, such as hallways, lounges, tea kitchens and elevator lobbies, to explore how technology can enhance informal encounters within these spaces. The project has placed several information kiosks around our campus building that deliver immediate access to digital information that is relevant and interesting to people nearby, and that foster a sense of community within large organizations by providing interactive public displays as a participatory shared medium of communication. Figure 1 shows one of those kiosks.

These information kiosks provide various services: they inform people of upcoming events such as lectures, show the time and weather information, and provide personnel directories and maps. They are conveniently available by corridors and elevators where people pass by. However, since these kiosks are not truly ubiquitous, and do not recognize the users, they cannot provide services such as reminding the user of a selected event or showing how to get there. Such services could be provided by an information appliance that can store and process information such as knowledge, facts, graphics, images, video, and sound.

A handheld information device, such a smartphone, can overcome many of these limitations. The device is personal and is easily carried around. The user can select a subset of interests out of the larger set of information available on a kiosk and download that to the device, and the device can remind of an upcoming event and provide associated information. The user can also take notes and associate new user-created information with those events, and share that information back with other users either directly and person-

ally, or via kiosk to everybody. The combination of a public kiosk and private handheld information device provides a good balance between public and private information, as well as potentially available vs. personally interesting information.

1.1 Conference information system

In addition to augmenting the usability of the kiosk by combining it with a smartphone we have another application that drives the design decisions of Kimono: an information system supporting researchers attending a conference. One or more kiosks will serve as a conference information center and likely reside in the lobby of the conference premises. It has the same information as the conference web site, and it would grab other relevant information from the event, hotels, travel to airport, tourist information of the conference location, etc. Changes in the scheduling of talks, lunches, dinners will be posted on the kiosk.

The conference participants can use the large touch panel display to select a personalized program, choose which of the parallel tracks seem most interesting, which booths to visit if a trade show is part of the conference, and so on. Only information that is selected to be of interest is then transferred to a handset (along with a client that knows how to communicate with kiosk if one is not present already). The data is displayed on the mobile device differently, taking its limited I/O capabilities into account [Metso et al. 2001]. On the handset the program can alert the user about the next event, when it starts and where it is, possibly even showing a path between the assumed current location and the target.

The conference attendee can record notes either as text, audio, video clips or still images, and associate those with events or items such as other persons attending a conference or a particular paper or poster presented at the conference. Votes for the best paper, etc., can be entered either directly to the kiosk or on the handheld device. People can exchange notes or virtual business cards directly between two handsets, or post some of their notes and votes on the kiosk. If any of the information previously downloaded from the kiosk have changed, the updates will be copied to the handset when next synchronizing with the kiosk. The architecture and most of the functionality of such a system has been implemented and is based on the ideas presented in this paper.

1.2 Previous work

The original OK-net kiosk [Van Kleek 2003a] was preceded by many other projects on information kiosks. The first widely available public kiosks were the bank ATMs already in 1970's. Morris *et al.* [1995] discuss the selection of components for constructing kiosk systems. The Digital Smart Kiosks Project [Christian and Avery 1998] studied how kiosks can be made more approachable by using animated avatars that would watch and summon users passing by. There has been a recent trend to concentrate on multimodal interfaces for a kiosk [Cassell et al. 2002; Johnston and Bangalore 2004]. Mobile devices with localization support have been studied for local services [Ojala et al. 2003]. Greenberg *et al.* [1999] studied how private mobile devices and public large displays can be used to facilitate computer supported cooperative work.

Our focus is on extending an information kiosk so that selected information can be downloaded onto a mobile de-
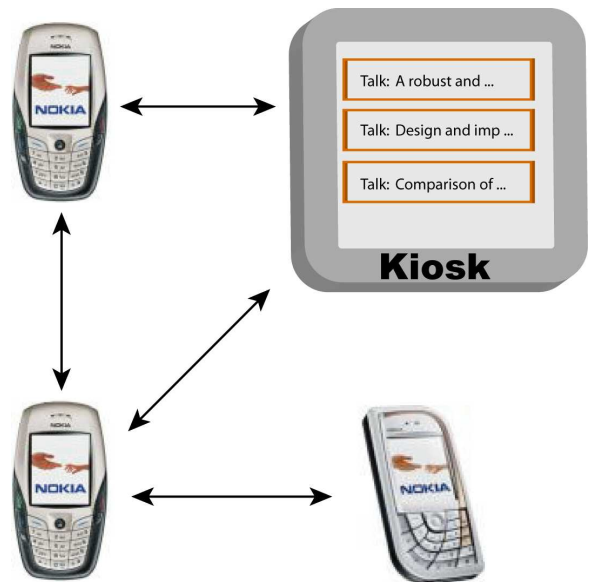

Figure 2: Information in the Kimono system flows both between handsets and kiosks as well as between handsets and other handsets.

vice, the mobile device can guide and remind the user of upcoming events, more information can be created and associated with existing information, and the new information can be shared with other mobile devices or the kiosk. This is illustrated in Figure 2.

1.3 Structure of the paper

We first give an overview of the kiosk component. We then describe both the architecture and implementation of the information exchange and processing system that runs on the mobile device and the kiosk, and briefly describe how the system is used in practice. We then describe planned extensions to the current system and conclude the paper.

2 Design Specification

Kimono consists of two different types of devices: kiosks and handsets, although many other devices can be part of Kimono as well. The main distinctions are in the user interface capabilities and whether a device is a public depository of information or a private subset of that information, with additional user-created data. The following describes the role of the kiosk as an independent information distribution system, and then the mobile devices that augment the kiosks. This section focuses on the required functionality and higher level system description, while the following section describes some of the implementation details.

2.1 Kiosk Functionality

The kiosk resembles a standard desktop computer built into a small wooden housing. Inside is a Pentium 4 processor, 512 MB RAM, a 40 GB hard disk, and a 3M touch panel display. USB Bluetooth and 802.11b adapters provide connectivity to the Internet and nearby devices. An array microphone and a speaker are used for preliminary speech interfaces. A keyboard appeared in early versions of the kiosk, but was

Figure 3: The new Kimono kiosk emphasizes an intuitive user interface and the ability to exchange information with a variety of mobile devices.

removed to encourage development of more intuitive, user-centric interfaces—users are now expected to interact with the kiosk using the touch screen. The kiosk runs Debian Linux 3.1 (Sarge) and to be secure, all ports and external services are closed except for connections to a central database and for Bluetooth communication. In other words, the kiosk is an information appliance and does not need constant maintenance, nor does it require constant security updates. Figure 3 shows one kiosk currently being used at our institution. To maintain compatibility with code that runs on the smartphone, the software for the kiosk is written in the Python programming language, using the GTK windowing toolkit and a C extension module that provides Bluetooth functionality in Python.

Kiosks play a central role in aggregating, displaying, and distributing information. A backend server automatically harvests information from web sites, e-mail lists, and other data feeds, which would then be aggregated and displayed to users [Van Kleek 2003b]. The current version of the system supports a network of kiosks and mobile devices, where information can travel in any direction without having to pass through a central authority.

Information Information is primarily obtained in three ways. First, data can be manually entered into the system by a trained operator. Although this is the simplest method, it is also the most tedious and labor intensive. Second, the kiosk can be configured to periodically check sources on an intranet or the Internet for new information. The kiosks have been configured to harvest information from the department's event database, the University's general event database, and local weather forecasts. When idle, the kiosk periodically cycles through short descriptions of upcoming talks, guest lectures, campus-wide events, and weather information. Although configuring data sources requires technical knowledge, data sources only need to be configured once and require virtually no maintenance. Since we want the kiosk to behave like a dedicated, stand-alone appliance, updates are made in the background and are stored to its large local disk and do not interrupt or affect the user interface. If network connectivity is unreliable or unavailable, the updates are postponed until data sources are once again accessible. Finally, the kiosk may obtain new information from mobile devices passing through the area, see Section 2.2.

Usage Models The kiosks are placed next to elevators on several different floors of the building, where users will typically browse the available content while waiting for the elevator. The kiosk is designed to be a quick information access station. When there is no active user, it randomly displays information, sorted by relevance to the immediate future. Data such as local weather information and scheduled talks about to happen will be displayed more frequently, events that won't happen for a couple days will be shown less often, and events that have already ended or have no time dependence will not be displayed at all. To avoid cluttering the display, and to increase the visibility from a distance, only one item is shown at a time.

A lab-wide directory has been integrated with the kiosks, and it allows users to find information about current lab members, as well as call their offices with the kiosk using a software voice-over-IP client that connects to the MIT phone system. Although no formal user studies have been completed, feedback is generally positive and lab members note that they find the information presented on the kiosk useful and clear.

A user begins interacting with the kiosk by touching the display, which will then show a list of data items centered on the current time. The user can browse this list, use an on-screen keyboard or a speech recognition system to search for an event or select a set of events. In the browse and search result views data items are abbreviated to fit in a small box, and tapping on a specific item will show it in greater detail along with summaries of related data items.

One or more selected events or objects can be marked as being interesting or uninteresting. A user can draw a check to indicate that the selections are interesting or draw an "X" to indicate they are uninteresting. We find such gestures faster and easier than the usual clicking of radio buttons, especially since it is not clear how to associate a radio button with a dynamically chosen set of events or objects. We use the term event generically, as a user may also select directory entries or directions to some destination as something that is of interest. Interesting events can be sent to the user's handset. Uninteresting events will never be sent, while unmarked events are sent at the discretion of the system (and based on the available space on the handset). Events contain a set of components, such as time, place, abstract, and perhaps even the slides of the talk. Which components are sent is described in Section 3.1.

2.2 Mobile device functionality

The kiosk has proven itself useful for the automatic collection and display of information to its users, but humans have very limited short term memory and often lose track of time and place. The data provided by the kiosk is not easily accessed when away from the kiosk, and users wanting to retain such information do not have many options. Early versions of the kiosk provided the ability to send emails or SMS text messages to users, but this is not a scalable solution. Sending an SMS to a user's cell phone is sufficient to describe and remind the user of a single lecture, but is not a good way to provide a user with a full schedule of conference events along with abstracts and descriptions of the speakers.

The phone application Storing information in a user's handset solves this problem. The large screen of the kiosk facilitates selecting which subset of the available information is of interest and should be copied to the handset. The contents can then be viewed at the user's convenience, and are as organized and accessible as they are on the kiosk. Infor-

mation can be sent to a smartphone, PDA, laptop, or any other device that has Bluetooth and can run Python. Our target mobile platform was a Nokia Series 60 smartphone. As with the kiosk, all application software was written in Python, using the Python for Series 60 user interface widgets and a C extension module for access to the phone's Bluetooth resources. Figure 4 shows a phone running the Kimono software application.
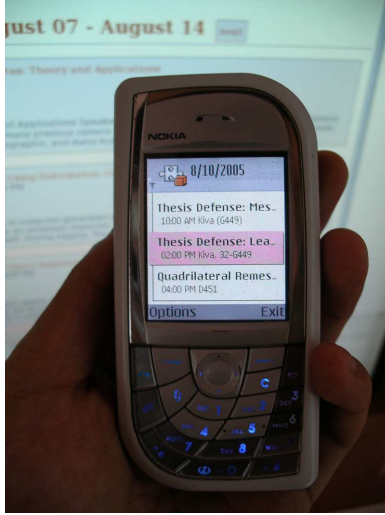

Figure 4: The Kimono smartphone application was first developed for Nokia Series 60 smartphones, one of which is shown here.

It is not expected that all users will have the Kimono application installed on their mobile phones. To obtain the application, a user standing at the kiosk initiates a software upload by pressing an indicated button, and the application is sent to his mobile phone via the Bluetooth OBEX push protocol. The user's phone prompts the user to accept a message and to install the Kimono application. Once installed, Kimono appears in the main list of programs available to the user.

Retrieving and displaying information The default behavior is to display a listing of data items centered on the current date and time. The user can browse this listing, moving back and forth in time, and select individual items for detailed display. On selection, an item is displayed along with a listing of related items. It is possible to mark an event as uninteresting, which is the same as deleting the event.

A Bluetooth connection is established and the data items transferred. Figure 5 shows a smartphone receiving data from a kiosk. The kiosk periodically performs a Bluetooth device inquiry to discover nearby Bluetooth devices. For convenience, a listing of these devices is shown in a small window for the user to select from when initiating a data transfer. Even if the user's mobile device is not discoverable by a Bluetooth device inquiry (e.g., for privacy reasons), the kiosk always is, so the user only needs to select the kiosk from his device to start the data transfer process. Moreover, any two devices running Kimono can exchange information without a kiosk; however, user action on each phone is required to setup a connection.

Creating new content In addition to viewing existing content, Kimono allows users to create their own content, and distribute it to other users either directly or via the kiosks.
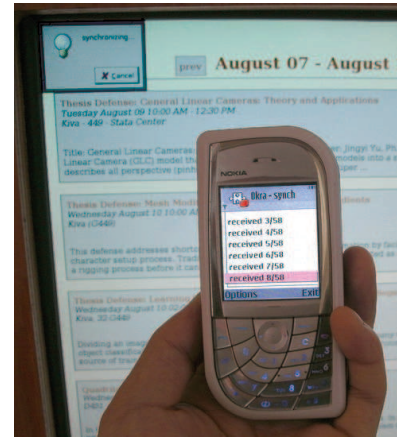

Figure 5: A Bluetooth data connection is used to transfer information between the smartphone and the kiosk.

Previously, a community member wanting to add content to the kiosks would need considerable knowledge of the system and its data sources. It is possible to provide users with an interface to do this at the kiosks, but the users would then be limited to adding textual content. Using a smartphone to transmit as well as receive data from the kiosk allows the user to distribute arbitrary data. New events can be created with basic information such as date, time, location, summary, and organizers. The user can also take pictures, record audio/video clips, and create text notes.

Newly generated content is added to the local database of information and is marked "public" by default. This generated content is automatically associated with the currently selected event. It is possible for the user to associate it with other events as well. The next time the smartphone connects to a kiosk, public objects will be exchanged provided they are associated with existing events (and in the case the connection is to another smartphone the event has been previously selected as interesting).

By allowing anyone to transmit content and information to the kiosk, where it will automatically be displayed, the kiosk closely resembles a public electronic bulletin board. For example, it is possible to define a week-long event entitled "Items for Sale." To post an item, a Kimono user needs to perform three steps: navigate to this event, take a picture of a paper advertisement for the item for sale, and synchronize with the kiosk. The picture of the advertisement will then be displayed when anyone looks at the event "Items for Sale" and anyone can get the information on their smartphone by expressing interest in this event.

Since posting new information is possible only from the database, the kiosk console, or through a local Bluetooth connection, the system serves only the local community and remotely spamming the system is not possible. It is also possible to protect areas of information with passwords, so that only some areas may be modified by anyone, or one may add, change, or delete information in some particular area.

3 Implementation

To maximize code portability and simplicity, we chose to have the same core engine code running on all Kimono devices, while implementing device specific user interfaces and
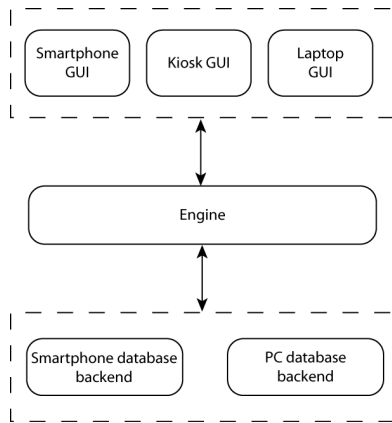
Figure 6: The modular design of the Kimono software allows us to use the same engine code on all platforms, and implement device specific modules when needed.

database backends. This is illustrated in Figure 6. To retrieve information, the user interface modules present a high-level set of attribute specifications to the engine, which then returns a set of objects for display and presentation. There is no direct interface between the user interface and the data storage modules. Instead, methods for information access are abstracted and placed in the engine module, which then interfaces with the database backends.

Modularizing the design of the Kimono system in this way greatly improved its reliability and allowed us to maintain the code base much more efficiently. Between the three obvious choices of C++, Java, and Python, we chose to use Python as the implementation language since Java is currently highly "sandboxed" on our smartphones, and C++ is significantly different on smartphones than on personal computers.

In the following we describe how data is represented, as well as the mechanisms for data display and transfer. The key concepts of association and policies affect the data organization, transfer, user interface, and display.

3.1 Mechanisms and Data Representation

There are four main concerns in the data representation used to store and organize information in the Kimono system. First is the correct interpretation and handling of different data types. Data passing through the system will typically be of several different types and formats, and applications must be able to display and present each item correctly. Second is the problem of efficiently distributing large amounts of data using relatively slow connections and storage devices. Third, organizing the information becomes more and more important as the volume of data grows, since the ability of an application to present an intuitive and simple interface to the user depends greatly on how well the data is organized. To address these concerns, metadata fields are created for each data object entering the Kimono system. Finally, we need to define a protocol for exchange of data and metadata between devices.

Handling multiple media types The first field is a type descriptor, which describes the data contents and how an application should interpret it. The type descriptor is similar to the MIME type widely used in Internet applications [Freed and Borenstein 1996]. The five main types used in the Kimono system are: *text, audio, video, image*, and *event*. The *event* type describes information that is heavily time-dependent, such as a meeting, weather forecast, lecture, or scheduled talk. It is the basic object used in our conference assistant application.

Efficiently storing and distributing data To avoid unnecessary copies, to save time, and to preserve bandwidth, data objects are assigned an *id* that is Kimono-wide unique. It is generated by the device that first created the data object using a timestamp and the Bluetooth address of the creating device and is analogous to an object handle or pointer. Two devices negotiating the transfer of a data object will use only its unique id until the actual time of transfer. The unique id is also used in organizing objects and creating meaningful relationships between them.

It is expected that all devices in the Kimono system will have a Bluetooth adapter and be able to generate unique ids in this manner. If the system evolves such that Bluetooth is no longer required, it is a simple matter to adopt another unique id generation method [Leach et al. 2005].

Organizing information - associations To organize individual data objects and construct meaningful relationships between them, data objects are grouped using named *associations*. A data object can be a member of one or many associations, each of which represents a set of data objects that are related in some way. Grouping data objects together in this manner is useful for several reasons. The user interface can use associations to display objects by their relevance to a topic or to another data object. Intuitively, it makes sense to present information that is closely related. However, the amount of information and how many different related objects are simultaneously presented depends on the display capabilities. Associations provide a way to scale the number of objects presented. Second, the synchronization protocols can use this information to transfer closely related objects together. Since bandwidth is a limiting factor when two devices communicate, and disconnections happen by people simply walking out of Bluetooth range, being able to exchange data objects in order of relevance ensures that they will have transferred the most important information first.

Associations can be given arbitrary names, which are intended to loosely describe how the objects in the association are related. For example, the association "HYPO2005" could contain all the events, images, text, and data related to the Hypothetical Conference 2005. There is no limit on the number of associations an object can be in, or the number of objects in an association.

Used in this manner, associations are useful for indirectly relating many objects. A special class of associations is used for stronger, more direct relationships. These associations are functionally identical to those described earlier, and differ only in how they are named. To express a direct relationship between objects, one object is added to the association "object-*uid*", where *uid* is the unique id of the other object. Creating associations this way is identical to constructing a directed graph where each data object is a node in the graph. Figure 7 illustrates several objects and the associations they have been organized into.

In most cases, the grouping of information into associations will be done manually by the user. In some obvious cases, objects can be automatically associated with high confidence. For example, if the user is viewing the details of an event listing on his mobile phone and takes a photograph, the photo is automatically associated with the event. Simple shortcuts like this have been implemented in the Kimono smart phone application, but more advanced methods

for robustly extracting semantic meaning from data sets are beyond the scope of the current project.
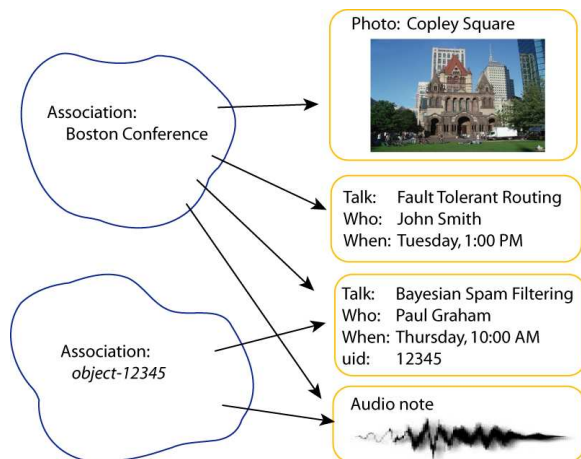

Figure 7: Example data objects and the two associations they are a part of.

Data Exchange The data exchange process begins with the establishment of a Bluetooth socket connection between two devices. The connection proceeds in five steps—*handshake, offering, request, object upload, disconnect.* An initial handshake is used to verify application and protocol versions, after which each device sends a list of unique ids corresponding to objects it is offering to send. Upon receiving the offering, a device will send a request list indicating the objects it does not already have a copy of. The offering device then sends the full data of each object requested along with metadata and associations that the objects belong to. When both devices are finished, the connection is terminated and the data transfer process is complete.

We use only simple techniques to avoid problems with coherency, updates, and redundancy. No attempt is made, for example, to merge changes made to a data object when two devices have concurrently made modifications to their own copy. Instead, the application takes the view that once a data object is stored locally, only the local user is permitted to make any changes. Future work may include the integration of techniques developed by projects such as PDIS [Rimey 2004], which do emphasize synchronization of widely replicated data sets. Events that are marked uninteresting or deleted are never exchanged, avoiding the annoyance of having data that was explicitly deleted reappear at the next synchronization with the kiosk.

3.2 Policy Specifications

Our desire for a simple user interface means minimizing the user's selection decisions. To this end, Kimono applications utilize a set of *policy specifications*. The policy specifications are intended to isolate and modularize many of the differences in display, user interface, data storage, data synchronization, and security. The policy specifications are not user-settable, except for a few cases where a simple and intuitive way for the user to change a policy was found. Three example policy specifications are described here.

The "fundamental object type" policy indicates to the user interface that a specific object type has greater importance than others. For the conference application, both the kiosk and smartphone define the event type as fundamental by default. On the smartphone, there is no way for the user to change this, but on the kiosk there is sufficient space for users to select any object type.

A second policy specification defines a default association behavior when creating objects. On the smartphone, for example, if the user is viewing an event and takes a photograph, the photograph is automatically associated with the event. This policy has no effect on the kiosk, where data creation by the user is not supported.

Third, the "interested in" policy defines default behaviors when users mark items as interesting or uninteresting. Objects explicitly tagged as interesting get precedence over those not marked at all (in terms of transfer and display), while objects tagged as not interesting are totally ignored. In the smartphone case, marking an item as uninteresting deletes it to save space.

4 Future Work

The architecture and implementation of Kimono is preliminary and can be extended in several ways, a few of which are underway.

Speech A speech recognition component has been incorporated into our kiosk system to allow a user to search the event database via spoken request. It is based on the Galaxy system [Zue et al. 2000], an architecture for creating natural language dialogs and is designed to be speaker-independent and robust against noise and speech impediments. We are planning to incorporate speech-based dialogue system to Kimono as well.

Video Another potential source of information to a kiosk is camera. One way a camera can be made use of is to track the gaze of the kiosk user [Morency et al. 2002]. This can be used as either a virtual mouse for making selections, or simply as an indication of user's interest, to which the kiosk can react, e.g., by giving more information on an item. One can also use it as additional information for a speech-based UI. The camera can detect which of the persons in front of the kiosk is speaking, and try to infer whether he is speaking to another person or the kiosk [Siracusa et al. 2003].

Security The design of the Kimono system currently has no reliable authentication method. Devices are identified by their advertised Bluetooth address, but this is easily spoofed. Without encryption or authentication, it would be possible for someone to plant a Bluetooth enabled device next to a kiosk that spoofs the kiosk's Bluetooth address. Such a device could perform a man-in-the-middle attack and send a malicious program to a user instead of the Kimono software application. To avoid this, the Kimono system could use the secret PIN facilities of Bluetooth to negotiate an encrypted channel with each mobile device. When a user first instructs the kiosk to upload the Kimono software application to his handset, the kiosk could randomly generate and display a PIN that the user must then manually enter into the handset. The two devices would negotiate a symmetric key, and the kiosk could then securely transfer the Kimono application to the handset. Since the PIN is never transmitted over the Bluetooth connection, a simple man-in-the-middle attack could not reliably guess it.

For the most part, there is no need to encrypt communication channels beyond the initial software upload. Since all the information transmitted during these data exchanges is public, eavesdropping is not an issue. In cases where sensitive information is exchanged, the Bluetooth PIN or an application-level encryption system would be sufficient.

In our initial implementation, objects were tagged as either public (the default) or private, but a more sophisticated approach is needed. A user may own several devices that run Kimono, such as a desktop, laptop, and smartphone. The user can type notes on the laptop and have them appear on the smartphone with the right associations but not be available for others to see. This can be done with the private attribute. Private data should be encrypted before being transferred, and the Bluetooth trusted device feature can be used to identify devices capable of sharing private data.

There are more complex relationships between devices owned by people. Some information, for example, can be shared with a colleague while other shared only with a spouse and yet another piece of information shared with the general public. We do not yet have a simple, user friendly way to specify such relationships or to automatically classify newly generated information.

Conferences seem to be an ideal testbed for Kimono. The information is highly structured and a conference kiosk is useful in and of itself. Many conference attendees already have smartphones and would welcome the features Kimono provides. It is also an ideal forum to test our conjecture that associations make selection and information interchange between attendees a natural and simple interface.

5   Conclusions

We have presented an architecture and initial implementation of an information kiosk system where information can be browsed and selected on a kiosk and selected information can be transferred to a mobile device where the information can be browsed at leisure, important events can remind the user about them, and the user can create more information and share that with other users. The concept of associations greatly simplifies the organization of data and synchronizations.

Kimono attempts to make the best use of informational devices in order to make the user interface as simple as possible. At first blush, a kiosk looks no different than a web page. However, since a kiosk is situated in a particular place in the physical world and there is a user standing in front of it, there is much context that can be exploited to simplify the user interface. For example, at a conference, the kiosk displays information relevant to the conference.

A smartphone has many distinct strengths and weaknesses when it comes to a user interface. The small screen and lack of full keyboard and mouse mean that only a small amount of information may be displayed at one time and the user should get to the important information with minimal actions. On the other hand, a smartphone is always with the user and can serve as a memory device, and warn or remind users about events that are about to happen. They remind the user about things that were marked as interesting, and can be used to record or make notes of things that are happening now for future reference.

The challenges of providing a simple user interface for presenting, organizing, and transferring information as well as exploiting context has been addressed by use of associations. A user need only express interest in some information displayed on the screen, in order to get it and its associated information onto his or her smartphone. New information can often be automatically associated with other relevant information and easily be made available to other interested parties. A mobile user equipped with only a smartphone has access to much information and the ability to generate new information; but with a small screen and keypad, these

functions must be easy to use. Kimono enables the user to comment on the action at hand with just two button presses: start and stop recording.

Acknowledgments

References

BLUETOOTH SPECIAL INTEREST GROUP. 2003. *Bluetooth Profile, Specification of the Bluetooth System, Version 1.2*, Nov.

CASSELL, J., STOCKY, T., BICKMORE, T., GAO, Y., NAKANO, Y., RYOKAI, K., TVERSKY, D., VAUCELLE, C., AND VILHJALMSSON, H. 2002. Mack: Media lab autonomous conversational kiosk. In *Proceedings of IMAG-INA '02*.

CHRISTIAN, A. D., AND AVERY, B. L. 1998. Digital smart kiosk project. In *Proceedings of the SIGCHI Conference*, ACM Press, 155–162.

FREED, N., AND BORENSTEIN, N. 1996. *Multipurpose Internet Mail Extensions (MIME) - Part Two: Media Types. RFC 2046*, Nov.

GREENBERG, S., BOYLE, M., AND LABERGE, J. 1999. Pdas and shared public displays: Making personal information public, and public information personal. *Personal Technologies 3*, 1 (Mar.), 54–64.

HUANG, A. 2005. *The Use of Bluetooth in Linux and Location Aware Computing*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

JOHNSTON, M., AND BANGALORE, S. 2004. Multimodal applications from mobile to kiosk. In *W3C Workshop on Multimodal Interaction*.

LEACH, P., MEALLING, M., AND SALZ, R. 2005. *A Universally Unique IDentifier (UUID) URN Namespace*.

METSO, M., KOIVISTO, A., AND SAUVOLA, J. 2001. A content model for the mobile adaptation of multimedia information. *The Journal of VLSI Signal Processing 29*, 1–2 (Aug.), 115–128.

MORENCY, L.-P., RAHIMI, A., CHECKA, N., AND DARRELL, T. 2002. Fast stereo-based head tracking for interactive environments. In *IEEE Conference on Automatic Face and Gesture Recognition*, 390–393.

MORRIS, G., SANDERS, T., GILMAN, A., ADELSON, S. J., AND SMITH, S. 1995. Kiosks: A technological overview. Tech. Rep. LA-UR-95-1672, Los Alamos National Laboratory.

OJALA, T., KORHONEN, J., AITTOLA, M., OLLILA, M., KOIVUMÄKI, T., TÄHTINEN, J., AND KARJALUOTO, H. 2003. Smartrotuaari - context-aware mobile multimedia services. In *Proceedings of 2nd International Conference on Mobile and Ubiquitous Multimedia*.

The oxygen kiosk network. `http://org.csail.mit.edu/oknet`.

RIMEY, K. 2004. Version headers for flexible synchronization and conflict resolution. Tech. Rep. 2004-3, Helsinki Institute for Information Technology.

RUDOLPH, L. 2001. Project oxygen: Pervasive, human-centric computing - an initial experience. In *Advanced Information Systems Engineering: 13th International Conference, CAiSE 2001*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 2068.

SIRACUSA, M., MORENCY, L.-P., WILSON, K., FISHER, J., AND DARRELL, T. 2003. A multimodal approach for determining speaker location and focus. In *International Conference on Multi-modal Interfaces*.

VAN KLEEK, M. 2003. *Intelligent Environments for Informal Public Spaces: the Ki/o Kiosk Platform*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

VAN KLEEK, M. 2003. k:info: An architecture for smart billboards for informal public spaces. In *Proceedings of the Fifth International Conference on Ubiquitous Computing*, 181–182.

VAN ROSSUM, G. 2005. *Python/C API Reference Manual*. Python Software Foundation.

WEISER, M. 1991. The computer for the 21st century. *Scientific American 265*, 3 (Sept.), 94–104.

ZUE, V., SENEFF, S., GLASS, J., POLIFRONI, J., PAO, C., HAZEN, T. J., AND HETHERINGTON, L. 2000. Jupiter: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing 8*, 1 (Jan.), 100–112.