Using Bluetooth Device Names to Support Interaction in Smart Environments

Nigel Davies, Adrian Friday, Peter Newman, Sarah Rutlidge and Oliver Storz
Computing Department
Lancaster University, UK
nigel@comp.lancs.ac.uk, adrian@comp.lancs.ac.uk, p.newman@lancaster.ac.uk,
rutlidge@comp.lancs.ac.uk, oliver@comp.lancs.ac.uk

ABSTRACT
An increasing trend in mobile and pervasive computing is the augmentation of everyday public spaces with local computation – leading to so called smart environments. However, there are no well accepted techniques for supporting spontaneous interaction between mobile users and these smart environments, though a wide range of techniques have been explored ranging from gesture recognition to downloading applications to a user’s phone. In this paper we explore an approach to supporting such interaction based on the use of Bluetooth Device (user-friendly) Names as a control channel between users’ mobile phones and computational resources in their local environment. Such an approach has many advantages over existing techniques though it is not without limitations. Our work focuses specifically on the use of Device Names to control and customize applications on large public displays in a campus environment. This paper describes our basic approach, a number of applications that we have constructed using this technique and the results of our evaluation work which has included a range of user studies and field trials. The paper concludes with an assessment of the viability of using our approach for interaction scenarios involving mobile users and computationally rich environments.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces

General Terms
Human Factors, Experimentation

Keywords
Public Displays, Bluetooth, Pervasive Computing

1. INTRODUCTION
An increasing trend in mobile and pervasive computing is the augmentation of everyday public spaces with local computation – leading to so called smart environments. This trend is fueled by the rapidly falling cost of hardware and opens up numerous research possibilities in areas such as cyber foraging where mobile users make use of local computational resources [2]. For the past four years we have been investigating the creation of smart spaces through the deployment of a research-focused public display network called e-Campus [26]. This network includes numerous large displays, each with associated computational, communications and sensing capabilities. A key question we have faced is how to support interaction between mobile users of these spaces and the installed public displays and associated resources. Previous research in this area (see section 6) has explored a wide range of techniques to support such interactions including the use of gesture recognition [28] for situations where users have no mobile devices with them through to sophisticated applications that are installed on a user’s laptop to facilitate screen sharing between a private and public display [9].

Our aim was to create an interaction method that enabled users to control and interact with content presented on the public displays. To this end we initially developed a system that enabled users to SMS requests for key pieces of content to be displayed on the screens. For example, a visitor to our campus could SMS the system to view a large map on a display. To provide further control of the system we also made available an application that could be downloaded to users’ phones (series 60 only). Using this application users could not only request content to be displayed but also issue simple requests, e.g. to highlight a specific location on a map.

Our experiences with these interaction techniques were disappointing. There is a natural, and justified, reluctance on the part of users to download new applications to their phone from potentially untrusted public locations. Furthermore, there is a significant overhead associated with developing high-quality applications for the wide range of phones in use on a typical campus environment. Similarly, interacting with a display using SMS presents a number of challenges. Firstly, users have to be made aware of the service and, crucially, how to parameterise the service for their current location. Using different phone numbers for each physical location creates additional work for the user as they can no longer store a single, trusted, phone number for a service. Using a unique identifier within the text message itself re-
Since changing the Device Name has a number of intuitively appealing properties:

- Low Barrier to Entry
  Since changing the Device Name is already part of most phone’s default functionality there is no need for users to download any new applications or learn how to use any new functions on their phone. Similarly, there is no need for users to carry a special device or tag to facilitate identification or interaction. There is also no requirement for users to register prior to using the service.

- Infrastructure Independent
  Since the user’s request is contained within the Bluetooth Device Name and the discovery of this Device Name is carried out locally within the smart space there is no need for the space to access any central database to interpret the query.

Free to Use
  It is clear to users that changing their Device Name does not incur any charges. Moreover, most users are more comfortable disclosing their Bluetooth Device Address and Device Name than they are their phone number - a disclosure which happens implicitly when they use SMS based services.

Supports Persistent Commands
  The use of Bluetooth Device Names as commands means that users can simply set their Device Name to an appropriate command and then issue that command automatically every time they enter a smart space. Commands can also be set ahead of time and then only issued when the user enters a smart space.

Supports Hands-Free Commands
  In addition to the notion of persistent commands we note that the actual transmission of a command requires no involvement from the user. Hence it is possible, for example, for a user to set up a command that is issued every time they walk into a smart space without requiring any interaction at all with the user.

Of course the approach is not without issues. Most obviously, it requires users to publicly broadcast their interaction requests; change the name of their device - a name to which they may be attached; remember interaction commands; and, tolerate delays when interacting with displays.

It should be noted that our target is supporting interaction with public displays and we are not suggesting the technique as a substitute for users accessing content for display on their own mobile devices. Example applications might include requesting a map of the local area, sharing photos or content amongst multiple viewers of the display or customizing a display as part of an artistic expression.

In the remainder of this paper we explore this approach in more detail. We begin by providing a brief overview of Bluetooth Device Names - both from a technical and a social perspective. We then describe our public display testbed that we have used to evaluate our approach. We then describe in detail the design, implementation and evaluation of a series of applications that support interaction through the use of Bluetooth Device Names. Finally, we present related work and our concluding remarks.

2. BLUETOOTH DEVICE NAMES

2.1 Technology
  Each Bluetooth device is addressable by a globally unique 48 bit Bluetooth Device Address that is assigned at the time of manufacture. In addition each device carries a textual identifier that can be assigned and modified by the user. These textual identifiers (called “Bluetooth Device Names” or “user-friendly names”) can be up to 248 bytes long when encoded using UTF-8 and are retrievable by other Bluetooth devices without prior authorisation by the user [4]. Bluetooth Device Names allow users to identify devices more easily when tasked with establishing a connection from one Bluetooth-enabled device to another. For example, a user

Figure 1: Interacting with the Map Application in the Nuffield Theatre

In this paper we explore an approach to supporting general interaction through the use of Bluetooth Device Names. Specifically, we allow users to make requests for content and to interact with applications by changing their devices’ (typically their phones’) names. This deceptively simple idea has a number of intuitively appealing properties:

Each Bluetooth device is addressable by a globally unique 48 bit Bluetooth Device Address that is assigned at the time of manufacture. In addition each device carries a textual identifier that can be assigned and modified by the user. These textual identifiers (called “Bluetooth Device Names” or “user-friendly names”) can be up to 248 bytes long when encoded using UTF-8 and are retrievable by other Bluetooth devices without prior authorisation by the user [4]. Bluetooth Device Names allow users to identify devices more easily when tasked with establishing a connection from one Bluetooth-enabled device to another. For example, a user

In this paper we explore an approach to supporting general interaction through the use of Bluetooth Device Names. Specifically, we allow users to make requests for content and to interact with applications by changing their devices’ (typically their phones’) names. This deceptively simple idea has a number of intuitively appealing properties:

- Low Barrier to Entry
  Since changing the Device Name is already part of most phone’s default functionality there is no need for users to download any new applications or learn how to use any new functions on their phone. Similarly, there is no need for users to carry a special device or tag to facilitate identification or interaction. There is also no requirement for users to register prior to using the service.

- Infrastructure Independent
  Since the user’s request is contained within the Bluetooth Device Name and the discovery of this Device Name is carried out locally within the smart space there is no need for the space to access any central database to interpret the query.

Free to Use
  It is clear to users that changing their Device Name does not incur any charges. Moreover, most users are more comfortable disclosing their Bluetooth Device Address and Device Name than they are their phone number - a disclosure which happens implicitly when they use SMS based services.

Supports Persistent Commands
  The use of Bluetooth Device Names as commands means that users can simply set their Device Name to an appropriate command and then issue that command automatically every time they enter a smart space. Commands can also be set ahead of time and then only issued when the user enters a smart space.

Supports Hands-Free Commands
  In addition to the notion of persistent commands we note that the actual transmission of a command requires no involvement from the user. Hence it is possible, for example, for a user to set up a command that is issued every time they walk into a smart space without requiring any interaction at all with the user.

Of course the approach is not without issues. Most obviously, it requires users to publicly broadcast their interaction requests; change the name of their device - a name to which they may be attached; remember interaction commands; and, tolerate delays when interacting with displays.

It should be noted that our target is supporting interaction with public displays and we are not suggesting the technique as a substitute for users accessing content for display on their own mobile devices. Example applications might include requesting a map of the local area, sharing photos or content amongst multiple viewers of the display or customizing a display as part of an artistic expression.

In the remainder of this paper we explore this approach in more detail. We begin by providing a brief overview of Bluetooth Device Names - both from a technical and a social perspective. We then describe our public display testbed that we have used to evaluate our approach. We then describe in detail the design, implementation and evaluation of a series of applications that support interaction through the use of Bluetooth Device Names. Finally, we present related work and our concluding remarks.

2. BLUETOOTH DEVICE NAMES

2.1 Technology
  Each Bluetooth device is addressable by a globally unique 48 bit Bluetooth Device Address that is assigned at the time of manufacture. In addition each device carries a textual identifier that can be assigned and modified by the user. These textual identifiers (called “Bluetooth Device Names” or “user-friendly names”) can be up to 248 bytes long when encoded using UTF-8 and are retrievable by other Bluetooth devices without prior authorisation by the user [4]. Bluetooth Device Names allow users to identify devices more easily when tasked with establishing a connection from one Bluetooth-enabled device to another. For example, a user
who is trying to synchronise their computer’s calendar with their phone is typically presented with a list of names of Bluetooth devices that are in range rather than a list of Bluetooth Device Addresses.

To retrieve the name of a remote Bluetooth device the requesting host requires knowledge of the remote device’s Bluetooth Device Address and an established logical link with the remote device. The address of the remote device can either be acquired through out-of-band means (e.g. cached from previous use or provided by the user) or through the means of device discovery, which in the context of Bluetooth is called an “inquiry”. In common with all radio communication in the context of Bluetooth, the inquiry procedure employs a frequency hopping scheme to combat potential interference in parts of the available frequency spectrum.

A device wishing to discover other Bluetooth devices repeatedly transmits inquiry packets on 32 dedicated inquiry channels. At least every 2.56 seconds each Bluetooth device selects one of these frequencies and listens for inquiry requests for a default of 11.25ms. If an inquiry request is detected, the device responds by sending an inquiry response on the same channel. The Bluetooth specification [4] states that in an error-free environment an inquiry should last at least 10.24 seconds if all devices in range are to be discovered.

Besides the Bluetooth Device Address of the responding device, the inquiry response packet also contains information about the device class (e.g. phone or computer), the clock offset of the device and the paging mode of the device (described below). The procedure for establishing a connection with a remote device is very similar to the procedure for discovering devices. A host wishing to establish a logical link with a remote device emits “page requests” on 32 dedicated paging channels. Each Bluetooth device is configured to periodically select one of these channels, tune into it for a default period of 11.25ms and respond to any incoming requests. Each device can implement one of three “paging modes” that define how frequently devices listen for page requests. Devices may either listen continuously, at least once every 1.28 seconds, or at least once every 2.56 seconds. The selection of the channel to listen to in each period is influenced by the device’s Bluetooth Device Address and by its own native clock. By using the clock information retrieved as part of the inquiry phase, a host wishing to establish a logical link with a remote device is able to predict which channels the targeted device is most likely going to listen to and can therefore target the page request to this set of channels in an attempt to speed up the connection procedure.

Once a device has received a page request, it responds with a page response, leading to the exchange of configuration parameters and finally the establishment of the logical link that can be used to retrieve the remote device’s Bluetooth Device Name.

### 2.2 The Use of Device Names

In 2005/2006 O’Neill et al. published the results of a study [18] that had captured and collected Bluetooth names of passers-by in three locations in the city of Bristol in the UK. This study showed that depending on the location between 58% and 88% of the captured Bluetooth Device Names had been user-defined.

In 2007 Kindberg and Jones conducted a series of follow-up interviews [12] aimed at investigating people’s use of Bluetooth Device Names on mobile phones more closely. The study found that besides the function of disambiguating between different devices, the names also provided various functions in the social contexts in which users were engaged. According to the study, the participants mainly used the Bluetooth capabilities of their mobile phones to exchange photos and music clips with friends. Besides names that reflected the identity of the owners (e.g. nicknames) and therefore enabled friends to form a mental connection between the devices and their owners, a significant percentage of the names used by study participants described properties and aspirations of the phones’ owners. These statements, for example, included information about the owners’ backgrounds or their interests. Names were also found to serve as props for direct social interactions, both within the owners’ immediate groups of friends, but sometimes also with strangers. In fact some participants deliberately selected device names that would provoke curiosity and encourage such interactions when discovered by others. In this context Kindberg and Jones identified that during the Bluetooth discovery procedure Device Names would “spill” between social groups: a user using his phone to discover the phone of a friend for the purpose of exchanging media files will inevitably also be presented with the device names of other Bluetooth-enabled devices that are in range at the time.

### 3. THE E-CAMPUS SYSTEM

The e-Campus system is a network of public displays on the campus of Lancaster University in the UK designed to serve a dual role: as an infrastructure and testbed for local researchers and artists, and as a device for improving the experience of students, visitors and staff on campus. In line with Weiser’s vision of seamlessly embedding technology into our surrounding [29], e-Campus particularly aims at supporting and encouraging experimental, potentially multidisciplinary research in the fields of mobile and ubiquitous computing.

#### 3.1 Deployments

The e-Campus public display network currently comprises around 70 displays, including 40 electronic door displays installed outside offices and labs, 25 large 40” LCD displays that are mainly located in foyers of colleges, departments and lecture theatres, and 5 projected displays in various locations. With the exception of the doorplates all our deployments are driven by Mac Mini computers running Mac OS X.

Figure 2 shows the first set of e-Campus displays that were installed on the campus of Lancaster University. Three projected displays were installed side-by-side in the Underpass, an underground bus stop on campus. The deployment opened with a piece of context-sensitive art that was specifically commissioned for this space.

Figure 3 shows three 40" LCD displays that were installed side-by-side in the foyer of one of the main lecture theatres on campus. A large number of students pass through this space every day on the way to their lectures. Moreover, students arriving early for lectures use this space to linger and meet up with their friends.

The content shown on the e-Campus displays throughout the day is typically a mix of university-wide content (e.g. news released by the University Press Office) and content that is location-specific. In the case of the displays that were installed in the lecture theatre complex, this may, for exam-
ple, include changes to the schedule of lectures in that space or announcements for special lectures and seminars. In the case of displays that are installed in the foyers of colleges, location-specific content typically comprises announcement of events and activities within the respective colleges. Moreover, we have reserved a daily one-hour slot in the afternoon on all e-Campus displays for exhibiting artistic content.

3.2 Software Architecture

The e-Campus public display network is underpinned by our own software infrastructure. The aim of this software infrastructure is to create an open software system that enables content providers (i.e. primarily researchers, but also members of staff, and students) to create applications that can control the presentation of content on the e-Campus displays. The software infrastructure is divided into two parts: a low-level software infrastructure and API [25] that offers precise and immediate control over the presentation of content on the e-Campus displays; and a high-level, constraint-based scheduling service and API that offers an easy-to-use mechanism for requesting content to be shown based on the specification of constraints.

The low-level software infrastructure comprises a small number of management processes that are distributed across the display machines. The processes in our low-level software infrastructure are implemented in Python and communicate using a protocol that is layered on top of an Elvin [23] publish-subscribe event channel. At the heart of our abstractions is the concept of a Display. A Display primarily represents a possible outlet for content, but also provides an abstraction over the underlying display hardware, e.g. automatically taking care of powering on displays when content is shown. Application processes represent content items and are responsible for rendering content onto Displays.

Content providers control the presentation of content (Application processes) on e-Campus displays by implementing and executing Schedulers – Python programs that use our low-level scheduling API to control the life-cycle and visibility of Application processes on e-Campus Displays. Multiple different Schedulers may attempt to show content on the same set of Displays at the same time, and the arising conflicts are handled by the Display processes involved based on the priorities that are associated with Application processes. Higher-priority content is able to preempt a Display from a lower-priority content item.

By allowing content providers to implement their own Schedulers, we enable them to precisely control when a piece of content should be shown, on which displays it should be shown, and for how long it should be shown. Moreover, we do not restrict the criteria used to influence these decisions, or the types of interaction devices or context sources that content providers may wish to interface with as part of this process. However, we also acknowledge that this flexibility comes at a price – programming skills are required to implement a Scheduler. To ease this burden we also support a high level API to the e-Campus system.

The high-level constraint-based scheduling service acts as a system-wide Scheduler that shows content on behalf of content providers. The service is executed on one of our servers in the e-Campus public display network. Content providers request content to be shown by using the high-level API to manage Playlist Request entries in a centralised repository. A Playlist represents a description of how a set of content items should be orchestrated on a set of Displays. Request entries that are associated with a Playlist specify when that Playlist should be shown. Once the constraints associated with a Request can be satisfied, the high-level scheduling service uses the low-level scheduling API to show the associated Playlist on the targeted displays. We attempted to make the high-level API as language and platform independent as possible and hence the high-level API is fully HTTP-based, i.e. users invoke API operations by constructing HTTP GET requests.

4. PROTOTYPE SYSTEM

4.1 Architecture

Our prototype Bluetooth Device Name system comprises a Bluetooth scanner on each node, a PHP application that generates web pages in response to user queries and an e-Campus Scheduler application that shows the generated content on the relevant screen (see figure 4). Communications between the components is carried out using the Elvin event channel.

Each Bluetooth scanner continuously discovers Bluetooth devices in its vicinity and resolves the names of discovered
devices. The scanners use Elvin events to communicate information about the presence of Bluetooth devices to a system-wide Scheduler. The Scheduler continuously evaluates the received information, parses the provided Bluetooth Device Names to identify valid queries and uses the low-level scheduling API of our e-Campus software infrastructure to show the requested content on the targeted public display. The content itself is generated by a PHP application and consists of an HTML frame set into which the requested content is embedded.

Information about the requested application type (see section 4.2) and the query parameters are communicated to the PHP application using a shared database. The database is also used to communicate information about the play-out status back to the Scheduler that uses this information to, for example, decide when to remove the content again from the targeted public display.

In order to reduce the time that users of the Bluetooth Device Name system have to wait for their content to appear on the public displays, the content requested via Device Names is assigned a priority that is higher than the priority of ordinary day-to-day content. This means that content requested using Bluetooth Device Names is able to preempt displays from the running program of content that is shown by default in the display network. If at any point more than one valid Bluetooth-based request for content exists for a single public display then these requests are queued by the Scheduler and processed by prioritising requests originating from phones that have been served least recently. We do not attempt to prioritize requests if the same content is requested by multiple people though the system could easily be extended to do so. If a phone whose request has been queued leaves the vicinity of a public display before its request has been served, then the Scheduler removes the corresponding entry from the request queue.

In order to reduce the time that users of the Bluetooth Device Name system have to wait for their content to appear on the public displays, the content requested via Device Names is assigned a priority that is higher than the priority of ordinary day-to-day content. This means that content requested using Bluetooth Device Names is able to preempt displays from the running program of content that is shown by default in the display network. If at any point more than one valid Bluetooth-based request for content exists for a single public display then these requests are queued by the Scheduler and processed by prioritising requests originating from phones that have been served least recently. We do not attempt to prioritize requests if the same content is requested by multiple people though the system could easily be extended to do so. If a phone whose request has been queued leaves the vicinity of a public display before its request has been served, then the Scheduler removes the corresponding entry from the request queue.

In order to reduce the time that users of the Bluetooth Device Name system have to wait for their content to appear on the public displays, the content requested via Device Names is assigned a priority that is higher than the priority of ordinary day-to-day content. This means that content requested using Bluetooth Device Names is able to preempt displays from the running program of content that is shown by default in the display network. If at any point more than one valid Bluetooth-based request for content exists for a single public display then these requests are queued by the Scheduler and processed by prioritising requests originating from phones that have been served least recently. We do not attempt to prioritize requests if the same content is requested by multiple people though the system could easily be extended to do so. If a phone whose request has been queued leaves the vicinity of a public display before its request has been served, then the Scheduler removes the corresponding entry from the request queue.

In order to reduce the time that users of the Bluetooth Device Name system have to wait for their content to appear on the public displays, the content requested via Device Names is assigned a priority that is higher than the priority of ordinary day-to-day content. This means that content requested using Bluetooth Device Names is able to preempt displays from the running program of content that is shown by default in the display network. If at any point more than one valid Bluetooth-based request for content exists for a single public display then these requests are queued by the Scheduler and processed by prioritising requests originating from phones that have been served least recently. We do not attempt to prioritize requests if the same content is requested by multiple people though the system could easily be extended to do so. If a phone whose request has been queued leaves the vicinity of a public display before its request has been served, then the Scheduler removes the corresponding entry from the request queue.

4.2 Applications and Control Interface

Using the architecture described above we have implemented a number of applications for use on our display network. Each of the applications is described briefly below. We enable people to access applications by changing the Device Name of their Bluetooth device. All of the commands to interact with our system have the same basic format, i.e. ec <service_name> <params>

The initial “ec” is simply used to denote that this is a command to the e-Campus system. The service name is one of map, flickr, youtube, google, tiny and juke as described below.

map: Interactive Map The first service we constructed was an interactive map service. If a scanner detects the device name “ec map” it will cause a map of campus to be displayed on its associated screen. This map includes a key in which buildings on campus are listed in alphabetical order along with an associated building number. Users can highlight a building on the map by appending the building number to their map request. For example, “ec map 51” would cause the map to be displayed and building number 51 (InfoLab) to be highlighted. We show the map for approximately 30 seconds.

flickr: View Photos from Flickr Users can access photos on Flickr by changing their device name to “ec flickr <search term>”. For example “ec flickr oranges” would cause photos retrieved using the search term “oranges” to be displayed. We sequence through the retrieved photos for approximately 180 seconds.

youtube: View Videos from YouTube Users can access YouTube videos by changing their device name to “ec youtube <search term>”. In this case we play the top result from querying YouTube with this search term. The video is played in its entirety.

google: View Search Results Users can access the page returned as the top search result from Google by changing their device name to “ec google <search term>”. This is equivalent to entering the search term on the Google home page and pressing the “I’m Feeling Lucky” button.

tiny: Generic Web Access We allow users to retrieve an arbitrary web page using the TinyURL service [27]. This enables users to access web pages without having to enter URLs which may well exceed the maximum length for device names. For example, if a user changes their device name to “ec tiny 2rn6g4” they will retrieve our Departmental home page. This is most likely to be used in the context of users accessing pages with a persistent query, e.g. for a news feed that they wish to see on each display they spend time in front of.

juke: Access Audio Juke Box We have implemented a simple music juke-box service. By changing their device name to “ec juke” users can bring up a list of music available at their location. By subsequently changing their device name to “ec juke <song id>” the selected music track will be added to the queue of songs to be played.

While Device Names are limited in length this does not appear to be a significant issue as the parameters typically used with the above services are fairly short and the TinyURL service can be used for accessing general, potentially much longer, URLs. The complexity of changing a device name varies by device but both our own research and that of O’Neill [18] suggests that most users have experience with the process. It is usually possible to update a device’s name while Bluetooth is active and this does not, in general, break device pairings.

In addition to these generic services we implemented a series of specific services for large-scale trials and these are discussed in the next section. Of course not all applications are suitable for deployment in all locations. For example,
some services would allow users to display potentially offensive content while others would cause sound to be played which may not be appropriate.

4.3 Presentation

An example of the application in operation can be seen in figure 5. The system’s display is composed of a frameset of four frames. The majority of the display is taken up by a single frame that is primarily intended to show the user’s requested content. The content takes one of the forms described in the previous section, displaying a slideshow of pictures, a video, an application or web page as appropriate. Should the user submit a Bluetooth application request that is incorrectly formatted, or a request that results in an error (e.g. no content can be found) then this frame is also used to show an appropriate informative message. In the event that no user requests are currently available, a large version of the e-Campus logo is displayed in this frame until the application is replaced with other content on the screen.

On the right hand side of the screen a second frame provides instructions to users and is always displayed when the application is running. Directly underneath the user instructions frame and consuming the remainder of the right-hand vertical column, a third frame informs users of the current state of received requests. At the top of this frame, the next item due to be shown on the display is listed, followed by a complete list of received requests. The list of received requests is ordered to reflect the sequence in which they will be shown on the display.

Finally, a fourth frame occupies a horizontal strip across the bottom of the display and provides a countdown timer informing users of the remaining time for which the current piece of content will be shown on the display. This frame also provides a notice to inform users that the unique identifier associated with their Bluetooth device will not be recorded. Maximum durations are enforced for the display of all media types.

5. EVALUATION

5.1 User Studies

5.1.1 Methodology

We conducted a formative user study on campus to gather feedback on users’ attitudes to our system and the use of Bluetooth Device Names for interaction. The study was conducted at two separate locations on campus - the foyer of the campus theatre and the foyer of one of the colleges. Both locations had operational e-Campus displays installed. Results were collected from a total of twenty-four participants selected using opportunity sampling from those occupying the space surrounding the display screens. Demographic data for the participants was not collected.

The study comprised of three stages. In the first stage, the participant was presented with an initial set of survey questions to establish the capabilities of the user’s mobile phone, their familiarity with these capabilities and their normal phone usage patterns (with a specific focus on the use of Bluetooth, SMS messaging and downloadable applications).

The second stage of the study involved a pair of tasks designed to familiarize the participant with the use of Bluetooth Device Names as an interaction method. In the first task the researcher demonstrated the system – talking the participant through their actions as they requested content using the Flickr service through modification of the Bluetooth Device Name associated with their mobile phone; the second task required the participant to modify a Device Name (either of their own, or the researcher’s, mobile phone) to request content using the Google service and a search term of their choice.

Following completion of the tasks, the third stage of the study involved a pair of tasks that they typically send more than nineteen SMS messages a week.

Approximately one third of the participants reported that they paid for all of their SMS messages, a further third that they paid for some of their SMS messages, and a further third that they did not pay for any of their messages. Of the 24 participants, all but one currently owned a mobile phone. Bluetooth was supported on 17 phones; of these 5 participants reported that their phone was currently in discoverable mode, 8 that it was not discoverable and 4 that they were unsure whether their phone was discoverable or not. 13 participants were familiar with the process of changing their Bluetooth Device Name whilst 4 reported that they were unsure how to achieve this task. The majority of those participants with Bluetooth (11 of the 17) had changed their Bluetooth Device Name with the majority citing recognition (for example, by friends or family) as their motivation.

In addition to the default software bundled with their mobile phone, 11 participants reported that they were aware that their phone supported the installation of additional applications, 4 that their phone did not support such applications and 8 that they were unsure whether their phone supported additional applications. Of the participants who were aware that their phone supported such applications, the majority currently had between one and three installed on their phone.

5.1.2 Results

Our results are based on 24 participants and represent a formative user study - based on the sample size we can not, of course, claim that the results generalize and a more comprehensive user study would be required to gain further insights into user behavior in this area. Of the 24 participants, all but one currently owned a mobile phone. Bluetooth was supported on 17 phones; of these 5 participants reported that their phone was currently in discoverable mode, 8 that it was not discoverable and 4 that they were unsure whether their phone was discoverable or not. 13 participants were familiar with the process of changing their Bluetooth Device Name whilst 4 reported that they were unsure how to achieve this task. The majority of those participants with Bluetooth (11 of the 17) had changed their Bluetooth Device Name with the majority citing recognition (for example, by friends or family) as their motivation.

In addition to the default software bundled with their mobile phone, 11 participants reported that they were aware that their phone supported the installation of additional applications, 4 that their phone did not support such applications and 8 that they were unsure whether their phone supported additional applications. Of the participants who were aware that their phone supported such applications, the majority currently had between one and three installed on their phone.

Over half of the participants with a mobile phone reported that they typically send more than nineteen SMS messages a week.
Following the demonstration and user task all but 1 user reported that the service was easy or moderate to use (see table 2). We note that this user did not have Bluetooth on their phone so may have been generally unfamiliar with the technology. Furthermore, all participants were successful in completion of the demonstration task with only a single error observed.

Despite the reported ease of use, participants were very divided when asked if they would be likely to use such a service with 30% of participants giving a neutral response and those remaining equally divided between positive and negative responses (see table 3). When asked why they would or would not be likely to use the service again, participants primary reasons for use of the service included those related to the usefulness of the information available (e.g. “Because it is easy to use and a good way of getting information”, “Simple, takes a few minutes but really good idea”), the entertainment provided through use of the service (e.g. “it’s fun”, “entertaining – useful”, “practical jokes in public” ) and the ‘coolness’ of the service itself (e.g. “...it’s pretty cool”, “It’s awesome”). Several saw it as a way of using the screen to display information targeted at others (e.g. “advertising events”, “It’s fun + I can shamelessly promote my society...”). Participants primary reason against using the service focussed on a perceived lack of value for the service (e.g.”It’s just something which is more bother than its worth”, “No need to use”, “Nothing to display”) though one person said they simply “never used Bluetooth”.

When presented with the three scenarios, the participants showed a significant preference for the map service over both Facebook and TinyURL; Facebook was the least popular service (see tables 4, 5, 6). The influence of location upon the potential likelihood of a participant using the map service (see Figure 6(a)) was considerably less than that upon the other two services (see Figure 6(b)), particularly when the participant was restricted to considering only unknown locations.

Only one participant raised security or privacy concerns. In this case the participant was worried that by disclosing their Device Name to the researcher they would be enabling that researcher to download and install malicious software on the device. The participant expressed concern that such an attack would be undetectable once their device name had been revealed and could allow other persons to control their device (e.g. making expensive telephone calls or subscribing them to unwanted services). This is, of course, not the case and demonstrates a lack of understanding of Bluetooth technology on the part of the participant.

When asked to compare the use of the Bluetooth Device Name system with other mobile phone based interaction methods participants generally favoured the use of Bluetooth names over the use of a downloadable application. Over 86% (19/22) of participants who responded reported
that they would be “less likely” or “much less likely” to use the described services if they were required to send SMS messages rather than alter their Bluetooth Device Name. Reasons for selection of Bluetooth over a downloadable application largely focussed on the additional time and effort participants associated with such applications (e.g. “too much hassle”, “Effort”, “More trouble than it’s worth”).

Participants also somewhat favoured the use of Bluetooth Device Names over SMS messaging as an interaction method but to a much lesser degree. Almost half of the participants who responded (10/22) said they would be neither more or less likely to use the services if they were required to send SMS messages rather than alter their Bluetooth Device Name, whilst a third (8/22) reported that this change would make them “less likely” or “much less likely” to use the services - the remainder (4/22) all reported that this change would make them more likely to use the service. Selection of one method over another was based on a range of factors including cost. Of the 14 responses describing why a participant would be more likely to use one method over another 6 were focussed on the cost – or lack of cost – associated with SMS messaging. This was particularly evident for users who paid for each SMS they sent (see figure 5.1.2). Additional factors included the extra effort required to create SMS messages (e.g. “more trouble to key in message”), difficulties associated with using Bluetooth (e.g. “dont like Bluetooth technical stuff”) and the familiarity of SMS messaging (e.g. “SMS messages are sent every day so quick and easy to send another”, “know how to send an SMS, never use Bluetooth”).

5.2 Trials

5.2.1 Visit day

Our first large scale trial of the system took place during two University “Visit Days”. During these days the University puts on a series of presentations and tours for prospective students and their families. Approximately 5000 people attended each of the visit days. On the first visit day we made available the following applications: a Campus Map (with dynamic highlighting of places-of-interest), Departmental information (such as open-day lectures and events) and a listing of important contact details which visitors may find useful for the day in question. We advertised the presence of the service on the screens themselves using a 1 minute advert shown every 5 minutes. The system was deployed in 3 locations, i.e. The Great Hall, the Faraday Building and Furness College.

Our experiences during the first visit day were generally disappointing. Despite the system functioning correctly and there being over 5000 people on campus we had 2 people actually use the system. In part we hypothesized that this was due to the abundance of information available on campus during the day. Indeed, the University employed numerous student helpers and printed significant quantities of literature (including maps!) to ensure that visitors had all the information they would require.

For the second visit day we decided to increase the number of services - targeting specifically information that was not being provided in other ways. This new information included a regional traffic report service (which provided local road travel news) and a rail-delays travel information service. We prepared a short questionnaire to allow us to quickly question the members of the general public visiting the University. A new advertisement was also created and the deployment locations remained the same.

As with the first visit day the uptake was very low (1 participant). Indeed, we were also unable to get an adequate response to our requests for visitors to answer the questions necessary to complete the questionnaire (4 responses). With hindsight it appears that the visitors to the campus were simply too busy to deal with additional demands on their time.

5.2.2 Finding a Language

Our second trial involved a piece of new art commissioned for the e-Campus system by Lancaster University’s Nuffield Theatre. The art work, entitled “Finding a Language” [5], was a collaboration between a video artist and poet and involved the creation of a sequence of seven new short videos that explored the notion of space within the context of the University campus. In addition to showing the videos as part of the regular schedule of e-Campus content we also decided to allow users to explicitly request the video using the Bluetooth Device Name system (using the command “ec litfest”). Since the videos had a natural ordering we designed the system so that it kept track of which videos a user had seen before and ensured that they saw each video in turn. Once the user had viewed all the videos the system would start again with the first video.

Our vision was of users setting their Bluetooth Device Name and then exploring the University campus - viewing a different video at each display. In practice the system did not perform as one would have expected. Specifically, the application highlighted the highly variable nature of our Bluetooth scanning infrastructure. At some locations users were detected while they were still a significant distance from a display. As a result the videos would start before users arrived at the display and hence users would miss the start of the film. Similarly, at some locations there was a very significant delay before users’ Bluetooth devices were detected and the videos started (see section 5.3 for a detailed discussion of this issue). In these cases users often assumed the screens were not working and walked past the display - missing the video in its entirety and, more crucially, preventing them from seeing the video at all since the system recorded them as having seen the video in question and consequently played the next video in sequence when the user was sighted again. As a result we would not consider this to have been a particularly successful trial of the technology and illustrates the difficulty in supporting “walk by” applications.

5.3 Performance Issues

A key factor that impacts on the potential usefulness of

<table>
<thead>
<tr>
<th>Payment Plan</th>
<th>Prefer SMS</th>
<th>No Preference</th>
<th>Prefer Bluetooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Some</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>All</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1: Table showing the preferred interaction method (Bluetooth or SMS) of users who pay for Some, All or None of their SMS messages
Table 2: Participant Results in Response to the Question ‘Overall how easy did you find this service to use?’ for the applications demonstrated

<table>
<thead>
<tr>
<th></th>
<th>Easy</th>
<th>Moderate</th>
<th>Difficult</th>
<th>Impossible</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users with Bluetooth</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Users without Bluetooth</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Participant Results in Response to the Question ‘How likely would you be to use this service?’ for the applications demonstrated

<table>
<thead>
<tr>
<th></th>
<th>Very Likely</th>
<th>Likely</th>
<th>Possible</th>
<th>Unlikely</th>
<th>Very Unlikely</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users with Bluetooth</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Users without Bluetooth</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4: Participant Results in Response to the Question ‘How likely would you be to use this service in the following locations?’ for the Facebook Scenario

<table>
<thead>
<tr>
<th></th>
<th>Very Likely</th>
<th>Likely</th>
<th>Possible</th>
<th>Unlikely</th>
<th>Very Unlikely</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A public area within the colleges</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>A public area next to a lecture theatre</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>The Nuffield Theatre/Great Hall</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>The Underpass</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>The library</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5: Participant Results in Response to the Question ‘How likely would you be to use this service in the following locations?’ for the TinyURL Scenario

<table>
<thead>
<tr>
<th></th>
<th>Very Likely</th>
<th>Likely</th>
<th>Possible</th>
<th>Unlikely</th>
<th>Very Unlikely</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A public area within this university</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A public area within a residential section of an area you did not know well</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>The Nuffield Theatre/Great Hall</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>The Underpass</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>The library</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 6: Participant Results in Response to the Question ‘How likely would you be to use this service in the following locations?’ for the Map Scenario

<table>
<thead>
<tr>
<th></th>
<th>Very Likely</th>
<th>Likely</th>
<th>Possible</th>
<th>Unlikely</th>
<th>Very Unlikely</th>
<th>No Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A cinema/theatre within an area you did not know well</td>
<td>3</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A bus/train station within an area you did not know well</td>
<td>5</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A public library within an area you did not know well</td>
<td>4</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 7: Table showing the time taken to determine a remote device’s name.

<table>
<thead>
<tr>
<th>Device</th>
<th>Successful Request</th>
<th>Unsuccessful request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia N95</td>
<td>0.652 (stdev 0.395) s</td>
<td>5.190 (stdev 0.015) s</td>
</tr>
</tbody>
</table>

Table 7: Table showing the time taken to determine a remote device’s name.

Our approach is the inherent delay associated with resolving Bluetooth device names. As others have observed (e.g. [18]), resolving device names can take a significant amount of time. In [18] O’Neill et al. suggest using multiple physical Bluetooth scanning devices (e.g. multiple Bluetooth USB dongles attached to a single machine) to address this problem. In this case one device would be used to carry out the inquiry part of the process and then a series of additional devices would, in parallel, be used to contact discovered devices in order to determine their device name. Clearly such an approach carries with it significant additional complexity - both in terms of the physical installation of multiple scanners within a smart space and in terms of software to control these parallel activities.

In analyzing the performance of Bluetooth device name resolution we note that the delay is essentially comprised of two distinct elements:-

**Inquiry Phase** This is the time for the Bluetooth device to scan for neighboring devices. By default this is set to 10.24 seconds and reducing this time increases the probability of failing to detect devices in the vicinity. The result of the inquiry phase is a set of Bluetooth Device Addresses representing the nearby devices.

**Name Resolution Phase** Once a device has been detected the scanner needs to determine the device’s name. This is achieved by forming a communications link to each device in turn and asking the device for its name.

In addition there is, of course, the time taken for the system to actually respond to the command. In the case of our public display system this delay is due to the time taken to fetch the appropriate content, instantiate an application on the display that is able to render the content and preempt any existing content of a lower priority (or wait until higher priority content has finished). In our system this would represent a delay of approximately 3.2 seconds. If the display is already showing content requested by another user then the new user’s request will be added to a queue of content to be displayed and this queue will be visible to the user.

Table 7 summarizes the results of our measurements for the name resolution phase. These timings were generated using an average of 60 device inquiry and name resolution cycles between a Mac Mini acting as a scanner and a Nokia N95 phone as the target device. We note in particular that the default time-out for this phase is 5.12 seconds. Thus there is a significant overhead associated with trying to determine the name of a device that has moved out of range of the scanner (or is no longer available).

Many APIs, including that provided by Apple [1], provide an operation to scan for devices (i.e. initiate an inquiry) that will in its default setting also resolve the names of the discovered devices as part of the same invocation. In such cases the scanner will first perform an inquiry and then iterate through the list of discovered devices establishing a link to each one in order to determine their Device Name. In an environment with significant mobility this is likely to lead to extremely slow name resolution as many of the devices that the scanner attempts to contact will already have left the coverage area of the scanner. Each such failed communications attempt will require the scanner to time-out the request - which takes roughly an order of magnitude longer than it takes to resolve a name successfully.

Our experiences of using the Apple API backs this up - in our trials during the visit day we were routinely unable to discover the names of all of the devices present near the display. Moreover, users often had to wait significant periods of time after issuing a request before their content was displayed.

To address this issue we have implemented a revised Bluetooth scanner component that enables us to separate the two phases of name resolution. Specifically, our new scanner enables us to carry out inquiries to determine the Bluetooth Device Addresses of nearby devices (without resolving their names) and then to selectively contact discovered devices in order to determine their device name. The scanner does not require any changes to the standard Bluetooth protocols. We propose to use this scanner to reduce the time taken to resolve device names by carefully selecting the Bluetooth Device Addresses we attempt to resolve in order to reduce the number of communication time-outs while minimizing the delay to users of the system.

Our algorithm is as follows. We carry out a cycle consisting of an inquiry phase and a name resolution phase. Each phase is allocated a fixed duration (inq_duration and res_duration respectively). During the inquiry phase we conduct a single inquiry lasting inq_duration - typically 10.24 seconds. The results of an inquiry phase are used to populate and update a table of know devices. Each entry in the table consists of a device MAC address, a device name (empty if unresolved), a count of the number of consecutive inquiries during which the device has been sighted (c_sightings) and a flag to indicate whether or not the device has been sighted during the most recent scan. We define a low and a high threshold for the number of consecutive sightings (low_sightings and high_sightings). Devices whose number of sightings lie below the low_sightings threshold are not contacted, i.e. we do not attempt to resolve their device name. Devices whose number of sightings are higher than the high_sightings threshold are added to a low priority queue and are contacted infrequently. The remainder of the devices, i.e. those devices whose number of sightings lie between the two thresholds are considered likely candidates to issue a command and are added to a high priority queue and contacted frequently.

Following each inquiry we update the table as shown in figure 7.

Once the inquiry phase has been completed and the table updated we enter the name resolution phase. During this we process entries from the two queues of devices. The ratio between the time we spend on the two queues is configurable but we always service at least some entries from the low-priority queue to avoid the possibility of devices that have been seen many times never being able to send commands. For each device whose name is resolved we update the table. Commands are passed on to the rest of the system to process accordingly. If a device’s name changes at any time then we reset its sightings count to 1 to ensure that after a short delay the device will reenter the high-priority queue.

The setting of inq_duration, res_duration, low_sightings
for each device in the table:
  set the sighted flag to zero
for each device returned by the inquiry:
  if this is a new device
    create a new table entry
    increment c_sightings
    if c_sightings > low_sightings
      if the c_sightings < high_sightings
        add to high priority queue
      else
        add to low priority queue
    for each device in the table:
      if the sighted flag is zero
        remove the device entry

Figure 7: Algorithm for Efficient Name Resolution

and high_sightings can have a major impact on the performance of the system. We suggest that inq_duration is left as the standard duration (10.24 sec) as this provides a very high likelihood of detecting all of the devices in the vicinity of the scanner. If inq_duration is reduced devices will be missed during this phase and as result will be removed from the table prematurely. Given the relatively short time needed to successfully resolve Device Names the value of res_duration and the weighting between servicing the high and low priority queues should be set to enable, in the majority of cases, all of the devices in the high priority queue and a fraction of those in the low priority queue to be resolved. Care should be taken not to select too high a value for res_duration otherwise users will experience a large delay between entering the range of a scanner and their device’s name being resolved. For a typical public space such as a foyer or waiting room a figure of 15 seconds would enable approximately 30 devices to be resolved each cycle. This is likely to be more than enough to cover new and recent arrivals plus a fairly large percentage of the more static devices.

6. RELATED WORK

6.1 Public Displays and Mobility

Over the past 20 years, public displays have provided the research community with a rich platform for experimentation and research into interactions between mobile users and public infrastructure.

The peripheral nature of public display systems makes them suitable for raising awareness and disseminating information to individuals and communities. For example, in 1996, Finney et al. created FLUMP (The Flexible Ubiquitous Monitor Project) [8], a public display system that showed personalised information to passers-by identified using an active badge system. Information that could be presented on the screen could include the number of new and old mail messages of the user, along with sender and subject line for each unread message, upcoming appointments, a cartoon of the day, or the opening status of a local coffee bar.

Mark Weiser’s vision of calm technology [30] also inspired researchers to explore the use of non-graphic, ambient displays to convey information, e.g. [31].

Researchers have created various public display systems that acted as social catalysts and aimed at assisting users in sparking off conversations with other users. These systems typically visualise information about individual users in an attempt to provide clues for conversations. For example, “GroupCast” [14], developed by McCarthy et al., was able to identify people in its vicinity using an infrared badge system and used this knowledge to display information that reflected the interests of users lingering in front of the displays.

In 2005 Kray et al. presented “GAUDI” (Grid of Autonomous Displays) [13], an application for using public displays as dynamic navigation signs. A central server was used to store geometric models of buildings, to calculate routes and to disseminate route information to individual displays. This route information was then further evaluated locally on each display, taking into account display-specific context information before being rendered as output.

Entertainment was the main focus of “MobiLenin” [22]. MobiLenin enabled users to use their mobile phones to influence music choice in a restaurant by voting for a particular track of a multi-track music video. A large projected public display was used to display the music video and the results of the voting phase.

Content selection was one of the topics investigated by Müller and Krüger [16]. The authors proposed a methodology for creating models, based on a combination of detecting and identifying users using Bluetooth scanners, gaze recognition techniques, and the use of user-contributed personal profiles. Müller and Krüger also described a public display testbed that they intend to use to implement and evaluate the proposed approach.

A whole range of different technologies are used in the context of public display systems to provide users with the ability to interact with public display content and applications through their mobile devices. For example:

**Based on mobile phone cameras:** a number of research prototypes have employed camera-based mobile phones as pointing devices. The solution developed by Ballagas et al. [3] processed optical flows in camera images to detect phone movement relative to the environment. Mitchell et al. [15] used cameras on mobile phones to capture and recognise visual codes that were shown on public displays.

**UIs rendered on users’ mobile devices:** the users of MobiLenin [22] interacted with the public display application through a custom-crafted Python application that was installed on their mobile phones. Information between the phones and the display application were exchanged using HTTP over GPRS. UIs rendered on PDAs enabled users to interact with Jukola [17], a public display system for democratising the music selection in bars and pubs.

**Activity and gesture recognition using environment-based cameras:** Vogel et al. [28] employed wearable reflective markers in combination with a camera to identify users and to recognise user gestures. The display prototype produced by Sawhney et al. [21] was equipped with a network camera whose feed was processed using image differencing and face detection algorithms. The prototype was able to detect whether users were present in front of the display, whether these users were moving or stationary, and roughly which direction people were looking into. In the context of the Interactive FogScreen [20] laser pointers were used as input devices, and these were tracked by external cameras.

**RFID tags and readers:** the LED clusters of the Hello.Wall
display \[19\] were equipped with short-range RFID tags. Users could use PDA-like devices that were equipped with RFID readers to select individual LED clusters.

Our work is complementary to much of this large volume of work on public displays and mobility. Our approach could be used to extent the customization capabilities of systems such as FLUMP or provide an appropriate way of configuring the navigation tools provided by Kray. Our approach to supporting interaction is significantly different to those that have been presented previously - we do not require the use of dedicated mobile applications, or non-standard RFID based devices. Moreover, we support unobtrusive interaction which is not possible with gesture based systems such as those developed by Vogel.

6.2 Using Bluetooth in Public Spaces

In 2006 O'Neill et al. described \[18\] their experiences with augmenting classic methods of observing the use of urban spaces (e.g. using gate counts) with technology in order to capture information about the digital realm as well. Specifically, the authors employed both mobile and static Bluetooth scanners to collect this data. In this context, the authors discuss the difficulties of constructing walk-past applications using Bluetooth scanners, caused by the tension that exists between the limitations of Bluetooth technology (i.e. long discovery durations) and the desire to reduce the range of Bluetooth cells to small physical locations and spaces. Fá-tah et al. took this approach even further \[7\]. The authors used a number of Bluetooth scanners in a city to trace and visualise people’s movements as an approach for analysing of the use of urban spaces.

The ability to identify and trace users based on the Bluetooth Device Addresses of the mobile phones they carry was used by Sharifi et al. \[24\] to investigate methods for maximising the exposure of advertisements that were shown on public displays. One particularly aim was to make sure that each advertisement was seen by as many people as possible while minimising cases where the same advertisement was shown repeatedly to the same person. Finally, Kern et al. investigated user views on the use of Bluetooth scans for the implicit generation of user profiles for advertising purposes \[11\].

The closest to our system is work conducted by José et al. \[10\] who explored the use of public displays and Bluetooth to support situated user-generated content. Bluetooth scanners were used to detect the presence of devices, whose device names were subsequently shown on an associated display. Users were able to enhance their on-screen representation by embedding tags in their device name that caused matching Flickr photos to be displayed next to the user’s representation on the display. An enhanced version of the system stored historical information about encountered tags and used this to generate content that reflected the perceived interests of those likely to view the display. José et al. observed a willingness of users to change their device names in order to use the system: 126 out of 650 devices changed their device name at least once during the six week trial period, compared to zero observed changes during a four week control period before the system was deployed. In contrast to José’s work we have focused on the use of device names as a general invocation and interaction mechanism for applications such as interactive maps and web queries.

7. CONCLUDING REMARKS

In this paper we have explored an approach for interaction between mobile devices and smart environments based on the use of Bluetooth Device Names. From our user studies and trials we can make a number of observations about the suitability of our approach. Firstly, we note that the original benefits we suggested in the Introduction have been confirmed by our investigations. The user responses to our demonstrations and questionnaires are in-line with the results published by Kindberg and Jones \[12\] and suggest that the majority of Bluetooth users know how to change their Bluetooth Device Name. Moreover, when asked detailed questions about using the applications we have created most people considered the system to be simple easy to use and preferable to alternatives such as application download and using SMS messages. Collectively this appears to validate our assertion that using Bluetooth Device Names presents a low barrier to entry for users. We have also shown that it is possible to construct the system without requiring central databases or registration and that persistent commands, free service provision and hands-free operation are all possible. These properties of our interaction approach can bring substantial benefits over existing systems. Our initial concerns relating to privacy or user’s reluctance to change their names do not appear to have been founded.

We do however note that we have not, to date, been able to develop or demonstrate compelling uses for the technology within the content of our public-display system. Of the three applications we discussed with users the Interactive Map application was viewed most favorably by potential users but even for this application it is not clear that a large number of users would actually invest the time to make use of the system. Based on our experiences working on the “Finding a Language” trial we suspect that there are many interesting applications that are likely to emerge from collaborations with artists who can devise new user experiences based on this technology.

Our experiences of developing and deploying applications also suggest to us that one of the conventional ways of thinking about Bluetooth as a means of customizing displays as users pass by is difficult to realize using Device Names. This is due to the inherent variations in coverage of Bluetooth, the need to detect users well in advance of them reaching displays and the delays associated with scanning for devices and resolving Device Names. We therefore do not believe this is an appropriate approach for engineering “walk-by” applications on public displays at this time.

As part of our work we have proposed a new algorithm for resolving Bluetooth Device Names that can be tailored to the mobility patterns of a particular space and offers significant advantages over conventional approaches.

When we think of generalizing our interaction approach to other smart environments we believe that there are a number of possibilities. For example, Bluetooth Device Names could be changed to express a users environmental preferences or to inform the environment of a user’s special needs. In the absence of any other device there is an obvious lack of back channel so care will need to be taken to provide appropriate feedback to users on the state of their requests but in many cases this will be entirely practical.

In conclusion we believe that using Bluetooth Device Names offers a very easy method of enabling large numbers of mobile users to exploit existing technology to carry out simple
application control and interactions with local infrastructure in a smart space.

8. ACKNOWLEDGEMENTS

We would like to acknowledge the many people who have worked on the e-campus system. We would also like to acknowledge Albrecht Schmidt and his colleagues and Marc Langheinrich for their insights during early discussions relating to this work. The first version of our system was developed and deployed while Nigel Davies was a Visiting Researcher at ETH Zurich in 2007. Finally, we would like to thank Andre Hesse for his work on the design and implementation of the Bluetooth Device Name application.

9. REFERENCES


