Learning Topologies of Situated Public Displays by Observing Implicit User Interactions

Jörg Müller and Antonio Krüger

University of Münster, Germany
{joerg.mueller, antonio.krueger}@uni-muenster.de

Abstract. In this paper we present a procedure to learn a topological model of Situated Public Displays from data of people traveling between these displays. This model encompasses the distance between different displays in seconds for different ways and/or different travel modes. It also shows how many people travel between displays in each direction. Thus, the model can be used to predict where and when people will appear next after showing up in front of one display. This can be used for example to create continuous 'shows' spanning multiple displays while people pass them. To create the model, we use Bluetooth connection data of mobile phones people carry, and employ the EM algorithm to estimate mean travel times for different paths people take.

Introduction

Imagine yourself strolling down a main road in your city center a hot summer afternoon. The streets are filled with digital advertising screens that show advertisements for a few tens of seconds each. You heard about personalized advertising, where people are identified and delivered individual advertisements, but you don't like that. You prefer to make use of your right to privacy and stay anonymous. As you look towards one display, you see an advertisement showing an attractive girl that states "Hot, isn't it?". You wonder what this is about, but continue walking. A few meters further, you see the same girl again that tells you "If there only was some cooling..." and now you realize how hot it really is and get curious. Some meters further, you see the same girl again telling you: "You're really lucky! Buy 'di Cultelli's' ice cream right here!". You turn your head and see that you stand right in front of an ice cream parlor. As you enjoy your ice cream you realize that you were not really identified by the advertisements but did only ride a 'green wave' of continuous advertisements. It is easy to provide such continuous 'shows' for the user if he can be identified. But because of users right to privacy, at least some users will always stay anonymous to advertising displays. Nevertheless, if a topological model of the displays is available that allows to estimate the probability that a user passes one display a certain time after he passed another, the following advertisements could be presented for certain confidence intervals of arrival probability. Thus, 'green waves' of continuous advertisement shows, related by a common design, could be created. If these are designed accordingly, like in the example, it would not matter much if the user misses one of the advertisements. In this paper, we focus on the problem of how such a topological model can be created. We implemented and evaluated our approach in our test environment presented in figure 1. Of course, such a topological model representing the relative locations of displays could be used for many more applications, for example deciding which advertisement to show depending on the environment in general.

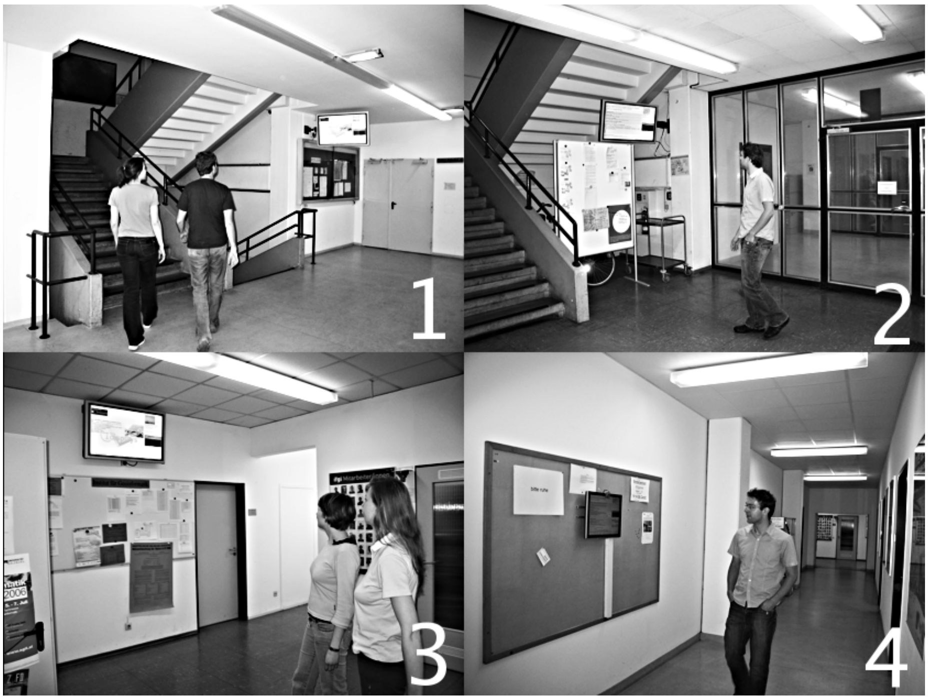

Fig. 1. The Situated Public Displays installed in our test environment at the University of Münster. Displays 1 and 2 are installed at the entrances of the building, while displays 3 and 4 are installed in hallways. The displays are used as an information system for the department and mainly show advertisements for talks, lectures, excursions etc. We have more than 10 sources of information from faculty and secretaries. More than 500 students use the system on a daily basis. The system is running for more than 15 months now, and over 340 advertisements have been posted. Each advertisement is shown on all displays for several weeks, thus there are always more advertisements available than fit on the displays. We select the advertisements that are shown in dependence on the location, time, and audience in front of the display. Most information is published as paper-based posters as well as on the displays, but the displays seem to be gradually replacing paper for urgent news.

The remainder of this paper is structured like follows. After summarizing some related work, we show how the problem fits into a more general context of self-configuring Situated Public Displays. Then we present how data obtained from people who can be identified can be used to create a topological model of the displays. This model contains the mean travel times between displays for different paths, the standard deviation and number of people that took that path. We identified people using Bluetooth sensors to identify their mobile phones. We present the algorithms we use together with the example data for our setting. Finally, we conclude and present some future work.

Related Work

This work builds on ideas that were introduced in the NearMe proximity estimation system [1]. NearMe estimates the proximity of different 802.11 wireless networking (Wi-Fi) enabled devices. When two devices share no common access point (long-range proximity), NearMe uses an estimation of the distance of access points the devices can connect to for estimating the distance of the devices. To achieve this, time-stamped Wi-Fi signatures are kept that store when the client was in reach of the different access points. The distance between two access points is then estimated as the minimum time difference between one signature with the first access point and another with the second access point. This is done for any pair of access points, resulting in a list of MAC address pairs and the minimum time any client was able to transition between them. Our approach is different from NearMe in two major ways. First, NearMe creates a full table of all pairs of access points together with a travel time estimate. Instead, we create a topological graph model that includes the information, which displays have to be passed to get from one display to another. Second, NearMe uses the minimum travel time ever observed to estimate the distance between access points. Our experiences have shown that because of frequent erroneous readings, this would lead to heavy underestimation of the distance between displays. We therefore use the Expectation-Maximization (EM) [2] algorithm to estimate the travel times between displays. Additionally, while NearMe uses Wi-Fi, we use Bluetooth, and while for NearMe a client software is necessary to collect the signatures, in our case the whole software runs on the displays. Predestination [3] is a system to predict the destination of car drivers from GPS data as the trip progresses. Similar to our system, it can be used to predict where a user will appear next. Location prediction is also a common problem in wireless networks, an overview can be found in [4]. In wireless networks, location prediction is used to make smooth handoffs between communication cells. Relate [5] is a system to sense the relative location of different devices with custom hardware. The spatial relations of a number of laptops on a table are sensed with ultrasonic transducers. A general overview of location systems for ubiquitous computing is provided by [6]. A more general approach to user modeling in ubiquitous computing is given by [7]. Here, Bluetooth sensors are used to detect users mobile phones and estimate users location and actions. In [8] Bluetooth sensors are employed to select advertisements not yet seen by the present audience in front of a public display. An overview of issues for self-configuration of Situated Public Displays, and a further description of Situated Public Displays as installed in Münster is provided in [9]. A collection of various works regarding Situated Public Displays is available in [10].

The Need for Self-Configuring Situated Public Displays

Regarding the adaptation of Situated Public Displays to their environment, we distinguish between the adaptation to the immediate situation (short-term adaptation) and adaptation to the general context (long-term adaptation). Long-term adaptation is especially important when there is not enough information available for short-term adaptation. For example, it could be impossible to identify each individual user in short-term. This could be due to technical limitations, but also due to legal issues or simply because users do not like to be identified. But long-term adaptation can nevertheless generate models that allow adapting to the general population of users, although the individual users can't be identified. In theory, these models could also be created manually, but in most cases this would not be realistic. There is enormous effort involved in creating such models and no central authority available to coordinate the creation across multiple display owners. Furthermore, the necessary knowledge that should be represented in the model might simply be unavailable. A continuously changing environment would require continuous updating of such models. Therefore, we follow the autonomic computing paradigm from IBM [11] and suggest that Situated Public Displays should be self-configuring.

One question that needs to be solved to enable such self-configuration is to determine the location of the display. The location could be represented either in absolute or relative coordinates. Location could also include models of paths people usually take, to predict where people would probably appear next. Absolute location models could be achieved from classical location systems, like GPS, together with maps of the environment. Relative location models could be created by observing the people passing the displays, like presented in the next section. The location of nearby Points of Interest (POI) would be needed to provide related information to the POIs. Such a model could be created by observing the position of people known to visit a POI before and during the event. Another question is how a model of the user population passing the display at certain days and times can be created and used to adapt information presentation to this population. Such a model could be created by observing the spatio-temporal behavior of people for whom profiles are available, for example via Bluetooth sensors. A model of which kinds of information certain people and groups are interested in is also necessary to enable such adaptation. This model could be created from interaction data, for example people copying events to their calendar. Alternatively, the display could observe which items people look at, for example via face detection with a camera installed at the display.

Learning the Topology of Situated Public Displays

To create a topological model of the displays, we follow a similar approach to that used in NearMe for long-range proximity. In NearMe, to estimate the distance between two access points, the minimum travel time ever observed was used. This can be problematic in case of frequent erroneous readings, for example trough ceilings and walls and due to numerous other error sources. In this case that approach is quite vulnerable to heavily underestimate the real distance. In our case, due to errors we sometimes have transit times of zero or one second between displays, so taking the minimum would clearly produce wrong results. Instead, we use the Expectation-Maximization (EM) algorithm [2]. This algorithm assumes that the different travel times are created by people taking different ways between the displays. It furthermore assumes that the travel times for each way are distributed according to the normal distribution. This is a strong simplification, because for example the normal distribution has non-zero probability for negative travel times while in reality negative travel times are impossible. Nevertheless, we show that this approach still produces quite good results.

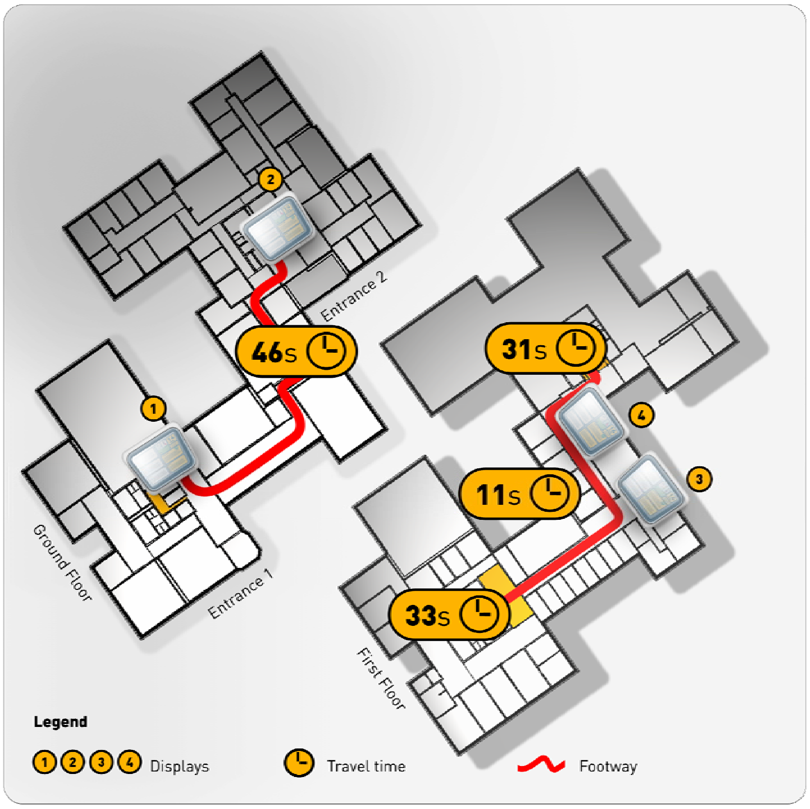

Fig. 2. A map of the test environment showing the locations of the different displays. The paths between the displays are shown together with the time needed to travel between the displays, as measured with a clock. These times are then estimated with the Expectation-Maximization algorithm.

The EM algorithm takes as input the set of travel times and the number of normal distributions (Gaussians) that should be assumed to have generated the data. It then adjusts the means and standard deviations of the Gaussians until it reaches a local maximum likelihood hypothesis for the parameters of the Gaussians. This approach has the benefit that it is quite insensitive to noise, because there is always one

Gaussian with a big standard deviation to accommodate all the noise. In addition, multiple paths and transportation modes, like walking, bicycle, car, bus etc. can be identified. The main drawback is that the number of Gaussians and thus the number of paths must be known in advance. In our case the data is generated with Bluetooth sensors. A Bluetooth sensor is a device that scans for nearby Bluetooth devices and reports the Bluetooth addresses of the devices found. Users can subscribe to a service that enables interaction with the displays via their mobile phone and thus agree that their data is logged by the system. Currently, most members of our work group and a number of students use this offer. Each display is equipped with a Bluetooth USB stick and runs software that continuously scans for nearby Bluetooth devices.

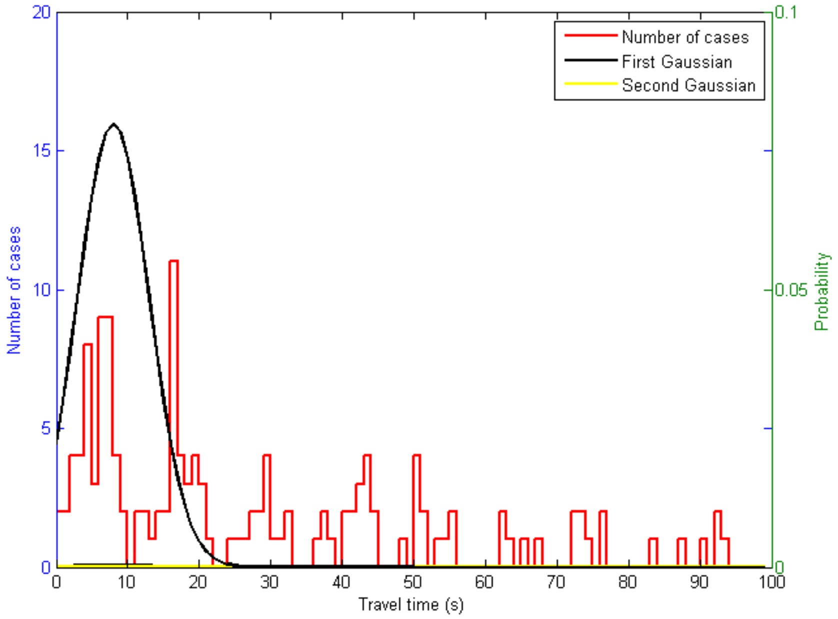

Fig. 3. Part of the travel times from display 1 to display 3, together with the EM estimations for two Gaussians. The first Gaussian (μ=8s, σ=5s) represents people who are assumed to have taken the direct way. The second Gaussian (μ=11284s, σ=63451s) represents the indirect ways and noise and is so flat that it is hardly visible. Thus, the algorithm assumes the path to take about 8 seconds to walk. Only travel times of up to 100 seconds are shown.

For all subscribed devices found it is checked whether they were present during the previous scan. If a device is found that was not present during the previous scan, a database entry is created that contains the Bluetooth MAC address, a timestamp, an “entered” flag and the name of the display. If a device that was present during the last scan is not found again, a database entry with a “left” flag is created. On our platform a scan takes 11 sec., so this is the temporal resolution with which timestamps are created. To calculate the individual travel times between displays, we cycle through all database entries with a “left” flag. From this entry, we look for the next entry with an “entered” flag, and store the time difference into the corresponding list for the pair of displays from which the entries were created. Thus, for each ordered pair of displays we obtain a set of travel times. For each ordered pair of displays we then execute the EM algorithm to estimate the mean travel times for the different paths between the displays.

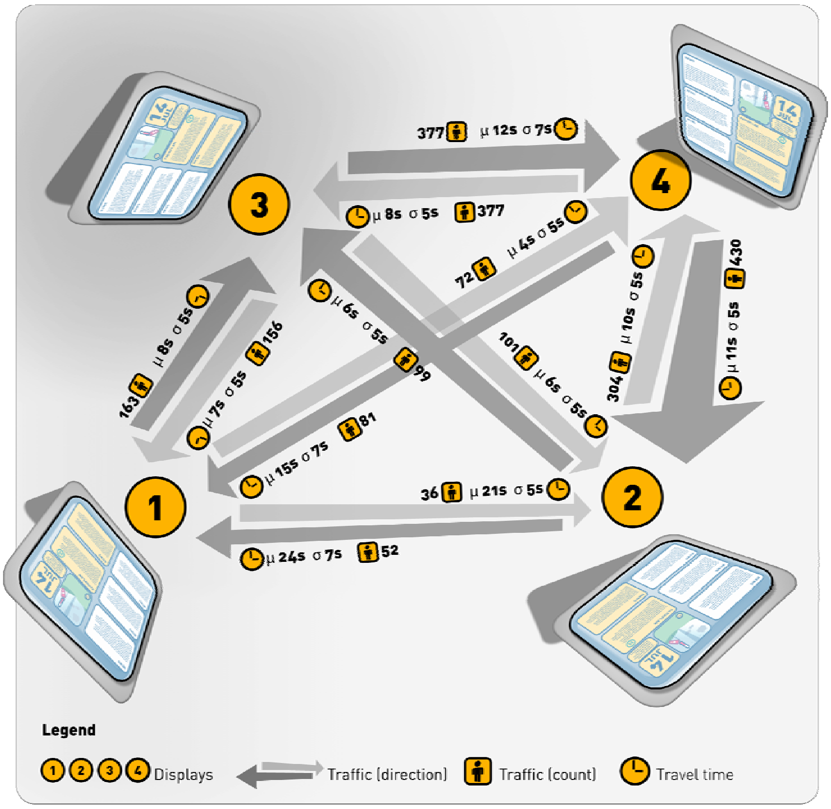

Fig. 4. The model of the displays in the test environment, as generated by the procedure presented. The model is a directed graph that contains as labels for each edge a) The mean time to walk from one display to the other b) The corresponding standard deviation and c) The people observed to have walked from the first to the second display. The model enables to predict where and when a person will probably show up next after having passed a certain display.

First we set the number of different paths we want to identify, in our case one per pair of displays. Then we add another Gaussian to accommodate the noise that occurs for example when someone passes a display, goes to vacation, and then passes another display when he comes back. The algorithm is then initialized with random hypotheses about the means and standard deviations of the Gaussians. In the Expectation step, for each of the instances the probability that it was generated by each of the Gaussians is calculated. In the Maximization step, new hypotheses about the means and standard deviations are built. This happens under the assumption that the instances belong to the Gaussians as calculated in the Expectation step. These two steps are repeated until the algorithm converges to a local maximum likelihood hypothesis. A graph showing the actual travel times from display 1 to display 3 together with the Gaussians that were calculated by the EM algorithm is depicted in figure 3. The data structure created by the procedure is a directed multigraph with labeled edges (see figure 4). Each display is represented as a node, and each Gaussian for a way between the displays is represented as an edge. Edges between two displays are labeled with a) How many people directly traversed from the first display to the second without passing another display in between and b) The expected mean and standard deviation of the travel time. Within four months of running the system we collected 92.827 timestamps. However, the majority of these timestamps were not actual traversals between different displays, but people sitting for example at their workplace and being repeatedly found and lost by the scanner.

From	To	Measured	EM	Error	Cases	PEBL	K700i	6230i	6310i
1	2	46	21	25	36	23*	23*	23	37*
2	1	48	24	24	52	x	38	30*	63*
1	3	37	8	29	166	x	-53	-7	-7
3	1	33	7	26	156	15	3	15	3
1	4	48	4	44	72	-	-	-	-
4	1	43	15	28	82	-	-	-	-
2	3	41	6	35	99	-	-	-	-
3	2	38	6	32	101	-	-	-	-
2	4	31	10	21	304	-12	-11	-11	-11
4	2	27	11	16	430	36	-21	-9	-30*
3	4	11	12	-1	380	16	-2	-4	-4
4	3	10	8	2	379	-7	-18	-18	-19

Table 1. Travel times between all combinations of displays in our building. The “Measured” column gives the times we measured with a clock, “EM” gives the estimation of our procedure for the distance, and “Error” is the difference between the two. “Cases” gives the total number of traversals our system observed, and the last four columns give the time difference between leaving the first Bluetooth cell and entering the next Bluetooth cell for different mobiles we carried while measuring with the clock. In case of negative times, the second cell was entered before the first one was left, impossible trajectories are marked with “-”. In case of “x” the device was not found by the display, and in cases of “*” the device was found by a third display that was not directly passed before being found by the second (target) display.

In total 2257 traversals between different displays were detected. Because within our building, between each pair of displays only one path is possible (see figure 2), we used for each ordered pair of displays two Gaussians. One of these was supposed to accommodate the data from people taking the direct way, and one to accommodate the noise. To create a ground truth, we measured the time needed to traverse between the displays by walking normally (see table 1). While we did this, we carried four mobile phones with us to generate comparison data for Bluetooth cell entry and leaving with different models. The data we generated for the travel times are shown in table 1. As can be seen in the table, there is a quite constant error of about 28 seconds except for the way from display 3 to display 4 and back. The mean error is 23.4 seconds. The error stems from four main sources. First of all, of course the Bluetooth cells are much bigger than only the immediate vicinity of the display. The Bluetooth signal is lost by the system only a number of seconds after the user started to walk away, and the second display receives a Bluetooth signal a number of seconds before the user actually arrives in front of the display. This sometimes even causes the cells to overlap, as with displays 3 and 4. Second, a scan can take place only every 11 seconds, so we have an error of up to ±11 sec. due to temporal resolution. Third, quite often a mobile phone is not found by a Bluetooth scan although it is in vicinity of the display. This causes an additional error of 11 sec. per scan where the phone is not found. Fourth, Bluetooth works through ceilings and walls. So, when a user passes next to a display on a different level or on the other side of a wall, he is often detected by the display. In our case, users traversing from display 1 to display 2 on the ground floor are regularly detected by displays 3 and 4 on the first floor (see figure 2). In table 1, these cases are depicted with a "*". To deal with the first error source, it would be sensible to add a constant offset of 23 seconds to the estimation of distances. The remaining error would then be within a few seconds, but that would have to be tested against a different set of displays in another setting. Because the error from the first three sources doesn't grow with the distance between displays, the bigger the actual distance, the smaller the impact of the error.

Conclusion & Future Work

In this paper we presented a procedure to learn a topological model of Situated Public Displays from data of people traversing between these displays. The model contains the mean travel time between Situated Public Displays for different ways and/or transportation modes. In addition, it shows how many people traverse between Situated Public Displays in each direction. This model enables to predict where and when people would probably appear next after showing up in front of a display. This enables us to present continuous 'shows' that span multiple displays while people pass them, regardless whether people can be identified, detected, or are invisible to the system. We implemented the procedure and tested it using four different displays. We showed that the procedure works for our case and presented four main sources of errors. We plan to add additional sensors to augment the model with further information. One idea is to use display-mounted cameras with blob detection to estimate in which direction a user walked after passing the display. This information could be used to generate arrows that guide the user to a particular location. Camera information could also be used to estimate whether a user actually passed a display and when exactly he did so. We also plan to work on the generated shows that span multiple displays, and want to further investigate how to generate such shows for people invisible to the system.

Acknowledgements We would like to thank all the people that contributed to this research: Patrick Olivier had the idea for continuous shows, Oliver Paczkowski and Oliver Meyer designed the Situated Public Display prototype. Dennis Wilmsmann implemented the Bluetooth Scanner, and Christophe Littfass created the illustrations.

References

1. Krumm, J., Hinckley, K.: The NearMe Wireless Proximity Server. In: *Proceedings of UbiComp 2004* Springer LNCS 3205, 283-300 (2004)
2. Mitchell, T.M.: Machine Learning. McGraw-Hill Science/Engineering/Math (1997)
3. Krumm, J., Horvitz, E.: Predestination: Inferring destinations from partial trajectories. In: *Proceedings of UbiComp 2006* Springer LNCS 4206, 243–260 (2006)
4. Cheng, C., Jain, R., van den Berg, E.: Location prediction algorithms for mobile wireless systems. In: *Handbook of Wireless Internet* CRC Press 245–263 (2003)
5. Kortuem, G., Kray, C., Gellersen, H.: Sensing and visualizing spatial relations of mobile devices. In: *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, New York, NY, USA, ACM Press 93–102 (2005)
6. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. In: IEEE Computer 34(8) 57–66 (2001)
7. Carmichael, D., Kay, J., Kummerfeld, B.: Consistent modelling of users, devices and sensors in a ubiquitous computing environment. In: *User Modeling and User-Adapted Interaction* 15(3-4) 197–234 (2005)
8. Payne, T., David, E., Jennings, N.R., Sharifi, M.: Auction mechanisms for efficient advertisement selection on public displays. In: *Proceedings of European Conference on Artificial Intelligence.* 285–289 (2006)
9. Müller, H.J., Krüger, A.: Towards situated public displays as multicast systems. In: *UbiqUM 2006 Workshop on Ubiquitous User Modeling, The 17th European Conference on Artificial Intelligence.* (2006)
10. O'Hara, K., Perry, M., Churchill, E., Russell, D. (Eds.): Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies. Kluwer Publishers (2003)
11. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. In: *IEEE Computer* 36(1) 41–50 (2003)