Tokenized Interaction Architecture

Boxian Dong, Bernhard Wally, Alois Ferscha

{boxian.dong, bernhard.wally}@researchstudio.at, ferscha@pervasive.jku.at

Abstract: Out-of-Home (OOH) advertising is currently advancing into a new era: “pervasive advertising” is one of the buzz words describing a soft change from the traditional push-only media to an interactive and bi-directional advertisement experience. While ambient media solutions try to conquer each and every piece of potential advertisement surfaces by various sorts of print media, another trend can be observed as well: OOH advertising is becoming smarter (at least from an electronics point of view). Digital signage products, SMS, Bluetooth interaction in city lights and talking billboards that react on the feedback of motion sensors—pervasive advertising is on its way. However, the development of a pervasive advertisement platform is beyond the scope of any single engineering or art discipline; instead it is a multidisciplinary effort including electronic engineering, computer science, art, cognitive psychology and even social sciences. Therefore, we introduce a *tokenized interaction architecture*; a foundation on top of low level issues that simplifies sub sequent implementation efforts by addressing interaction, context awareness and hardware heterogeneity, so to release designers of pervasive advertising campaigns from computer related challenges.

1 Introduction

New wireless communication technologies, the unceasing evolution of CPU power and efficiency as well as improvements in signal processing systems make the Weiser’s (1993) vision more and more realistic. Many implementations show the feasibility of systems where numerous computing devices are widely distributed throughout our natural environment but stay invisible to the human eye, operating in the background to support natural human behavior and interaction. At the same time, the advertisement industry is driving and expanding their market by permanently looking for and introducing new ways of advertising. Pervasive computing is a powerful force to advance and support the advertisement industry, because of its capabilities like personalization and ubiquity.

In our vision, it is necessary to introduce a relatively specific architecture to be commercially implemented, with the following key aspects: integrability into existing advertising platforms, hardware abstraction (release designers from technological burden) and a clear focus on innovative forms of advertisement. This paper first analyzes the features of pervasive advertisement systems and their implementations in Sec. 2 to sub sequentially abstract the requirements of such systems. Sec. 3 will first conclude the related work in pervasive platform research, and then introduce a tokenized interaction architecture, optimized for interactive advertising. Sec. 4 presents a proof-of-concept demonstration of the system, followed by conclusions and an outlook in Sec. 5.

2 Related Work

2.1 Advertising Technologies

Advertisements that adopt pervasive computing technologies have been discussed and researched for years—several pervasive systems have been built to deliver context aware ads at the right time, to the right person at the right place [RC02]. [Gra02] and [BK01] use mobile phones as the main channel to display advertisements and other information, in [SPD06] a system that collects audience statistics by exploiting Bluetooth to detect people's presence is presented. [SRA09] shows how to track customer information by profiles generated by users' Facebook accounts through Bluetooth and RFID, while [BLSC01] and [BL97] use the interactive digital home as platform for advertisement delivery. [BK01] and [MKHS08] generate advertisement with capabilities to let people interact implicitly, [SRA09] and [BL97] focus on explicit interaction. Besides those applications, there are also some public display based implementations, such as those from [MAMI07], [BPS+08] and [BFS+08], where the usage of public displays also inherently provides potential for advertisement applications.

Previous work indicates how methods and technologies of pervasive computing can be used in the advertisement market as well as in public information systems. Based on these findings the remainder of this section will first analyze the challenges of and possible solutions for pervasive advertisement and then conclude with the requirements of a universal architecture to be capable of supporting future advertisement implementations.

2.2 Context Frameworks

[SBHM02] solved the communication decoupling by using context-based message exchange (context-matching) and unifying the messages produced by the input/output devices (common interface). However, [SBHM02] doesn't decouple the encapsulating software from the device, and runs the interface support tools and the processing engine on the physical device. This increases the computing efforts on the physical device and therefore limits the choice of the technology of the end device. [VSdI06] and [CFJB05] release the end device from computational expensive processes by introducing the context-broker that produces user-agents that run on a server on behalf of the end devices. This approach makes it possible to use cheap devices in the system like simple sensors or wirelessly controlled power switches by leaving the computing work to the user-agent in the network. [VSdI06] also improved the system capabilities presented in [CFJB05]'s by decentralizing the user-agent: through the introduction of the orchestrator which scans, matches and controls the distributed user-agents. However, [VSdI06] did not distribute the orchestrator itself and implemented both the intelligence function and the matching function in the Orchestrator, limiting the scalability and maintainability of the system. [BDO+04] divided the ambient intelligence into three layers: environmental, reactive and deliberative. The environmental layer abstracts all low-level interaction (sensors, actuators), the reactive layer delivers quick responses (and forwards more complex situations to the deliberative layer) and the deliberative layer handles all the complex tasks. This approach reduced the complexity of the implementation by simplifying the functions of each module, and increased the maintainability of the overall system. [BCA+08] also decouples the communication from context-matching. The system uses a center arbitrator Adaptation Decision Engine (ADE) to manage the communications, and uses a distributed Adaptive Engine (AE) to manage the behavior and intelligence of the end devices. This approach masks the communication and underlying technology details to a maximum, and enables the higher level engineers to concentrate on their applications. However, the center arbitrator Adaptation Decision Engine depicts a bottle neck of the system. Another framework for "the internet of things" has been presented in [RDG+08], involving OSGi with extensions such as R-OSGi over Bluetooth or IEEE 802.15.4. The system proposed modularizes the system in that it maps each hardware device as a software service into a distributed platform with support for streaming data such as continuously measured sensor data. From a technical point of view, the so called "software fabric" is closely related to the system presented in this document, especially when it come to the mapping of hardware devices to software service instances. However, while the system presented in [RDG+08] is basically a distributed framework for transparent access to various devices over a unified Java platform, the focus of our platform is on the ad-hoc communication of mobile devices ("tokens") with the environment by exchanging a small set of information (a "profile"). In our architecture the (remote) transparent invocation of methods is not a basic requirement nor is it the core component to be implemented—we require a platform that is able to react on spontaneous implicit user interaction by adapting service behavior to the profile data read out. The system presented in [SK05] focuses on (tangible) user interaction and methods for multi modal, wearable, augmented reality (AR) based user interfaces. With a clear focus on explicit and immediate user interaction, the system is a framework for handling various input methods in a consistent manner to allow presenting AR scenes accordingly. Our envisioned use case, implicit user interaction in OOH advertisement settings, has no connection to AR technology or tangible user interaction at all. [DAS01] is an early example of a context aware computing platform developed for a smart conference setting. It aggregates various context data including location, user interest level, presentation content, etc. into a context-aware application that can among others support attendees of a conference in selecting workshops in a multi-track schedule based on the interests declared at the registration. Especially the people interests part is of relevance to our own architecture, as it gives an example of how interests (and their weighting) can be used to build an adaptive system.

To solve the challenge of versatility, we adopt the AE concept from [BCA+08], and the context-broker concepts from [VSdI06] and [CFJB05] to minimize the computation requirements at the end device. To solve the challenges of communication decoupling and scalability, we introduce a distributed context-based routing (similar to the context management in [SBHM02], or the AES in [BCA+08], but without intelligence).

3 Tokenized Interaction Architecture

The tokenized interaction architecture is designed to abstract technology related challenges like hardware heterogeneity, methods of communication and context matching in an interactive advertisement environment. We use the term "tokenized" because our concept is based on "tokens", the core functional units in our architecture. For the user a token could be a mobile phone, smart key fob, etc., while environmental tokens include sensor nodes, display adapters, etc (cf. Fig. 1).

Figure 1: Examples of possible user tokens (not to scale): (left and middle) a custom designed ZigBee platform and its housing (an actual car key), (right) a mobile phone communicating over Bluetooth or Wi-Fi.

3.1 Challenges to Advertising

Advertisements are all about influencing people, e.g. trying to make them purchase specific products or making them believe in the messages transferred by the ads. There are three important challenges to efficiently convince people: information delivery, information attractiveness and feedback, as concluded from [RC02]:

Information Delivery Traditional advertisements are distributed through mass media like TV, radio, billboards and newspapers in a unidirectional, broadcasting way. This works well to deliver information to the people who are interested, but at the cost of lavishing resources, because in order to guarantee the exposure the advertisements need to be delivered over and over again, and (as the main side effect of mass media) even to people that have no interest for certain topics. Pervasive advertising technologies can help to increase the efficiency by knowing the environment advertisements are delivered at [SRA09, BK01, BFS$^+$08].

Information Attractiveness Traditionally, there are two ways to make advertisements more attractive: either through adding emotional elements such as humor or consternation or through a focused adaption to an individual's personal interests. The first method has the advantage of universality (at the cost of impact) while the second method can only be applied to a certain group of people. In pervasive environments we have the chance to reach people in individual ways that rely on people's interests and can even introduce interactivity thus enabling more enchanting advertisements [BL97, BK01, MAMI07, MKHS08, BPS+08, BFS+08, SRA09].

Feedback The last very important but all too often neglected challenge of advertisement is user feedback. Feedback is not only the source for the advertiser to collect his revenues, but also for the advertisement and media agencies who design and realize the advertisement campaigns to name and number the results of their implementations in order to refine their strategies for future campaigns [SPD06].

3.2 Requirements for a Tokenized Interaction Architecture

Based on the discussion above, it is clear that a true pervasive advertisement architecture includes some kind of a context framework to understand and work with contextual data: when and where, who is around, what are their interests and what are they doing. Content needs to be presented with respect to the environment it is shown in: what kind of content, how to interact with the surrounding (choosing the right modality) and what kind of feedback can be read from the environment. To achieve this, we need an architecture which includes ambient awareness by sensors or "user profiles" present in the environment, which has the capability to choose the best output device based on the information it got from the environment and can record the environment changes for long term analysis and realtime user feedback. Therefore, the basic requirement of the platform is the capability of wiring different kinds of input, output and storage components together.

3.3 General Architecture

The architecture is mainly constructed by three components: one physical component (tokens) and two virtual components (token shadows and a virtual ecosystem). Tokens are the basic functional units in the system, including displays tokens, audio tokens, user tokens, sensor tokens, etc. For each token, there is a virtual representation residing on the servers which we call the "token shadow". Token shadows represent and abstract the heterogeneity of the physical tokens and their communication channels. The sum of the token shadows together with some core and extending services running on distributed servers form the virtual ecosystem. The computational heavy tasks such as supporting a unified interface for different hardware, context matching, communication decoupling and artificial intelligence can be realized in this ecosystem. Therefore, the token itself can be a very low-power device realized in any kind of technology. The ecosystem serves to decouple the communication, so that there is logically only communication inside the virtual ecosystem, while the shadows act as gateways to their physical representations if required (cf. Fig. 2). "Residential servers" act as hubs for ecosystem-side tokens, such as display or sensor tokens and are thus the physical representation of the gateways from the physical tokens to the ecosystem.

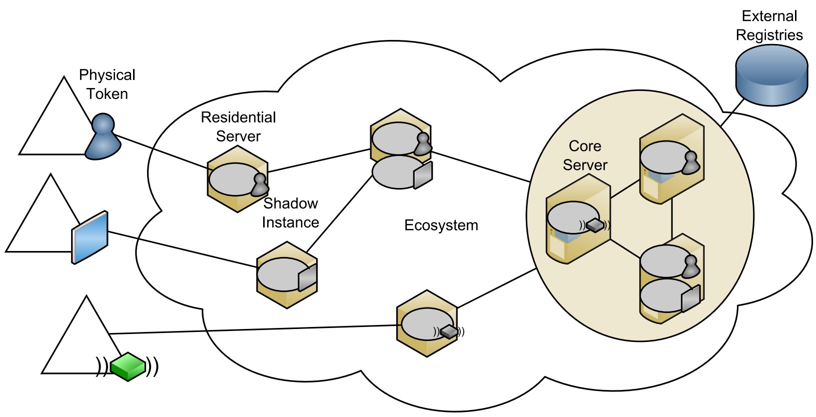

Figure 2: The tokenized interaction architecture: physical tokens and their specific technologies are abstracted by their respective token shadows that exist in a virtual ecosystem as instances of token shadows (those might be distributed within the ecosystem).

Fig. 2 demonstrates the general concept of the architecture. There are three tokens in this scenario: each a display, a user and a sensor token.

- The user token is typically mobile and connects to the system from various locations, while its shadow remains in the virtual ecosystem, allowing shadow instances to follow the user's token. The main purpose of the user token is to provide a user profile to the ecosystem which can be used to infer context information.
- The sensor token resembles an instance of an environmental sensor node that is able to capture environmental information (e.g. the amount of people around, lighting conditions, weather information).
- The display token uses contextual information it can gather from the virtual ecosystem to decide on the content to display. That includes information provided by sensor tokens and profiles injected into the ecosystem by user profiles.
- External registries depict content servers or other databases and computing systems that can be queried in order to get additional context information or to distribute content to display tokens.

As shown in Fig. 2, the ecosystem is to be implemented as a distributed system with a certain hierarchical approach, based on the underlying network topology: "core servers" act as the central entities that can be queried for the ecosystem's state. This mechanism is e.g. used by display tokens to receive information about their environment. Such calls are possibly routed through several residential servers that can act as a cache for the core servers, i.e. if they know the response to a query initiated by a shadow, they can respond accordingly without forwarding the request to the core servers. This mechanism is especially designed for shopping mall scenarios, where the various residential servers might be interconnected over a wireless communication technology like ZigBee, without direct access to the Internet. A special residential server would then act as the gateway of the sensor network to the Internet, thus each query that is targeted to the core servers passes this gateway. By implementing a cache of the core servers on this gateway, multiple requests for the same information could be streamlined to only one request that is sub sequentially delivered to the requesting shadows.

3.4 Software Stacks

The architecture has three software components: the software on the physical token (firmware), the software on the core and residential serves that makes up the virtual ecosystem and the software depicting the token shadows. The latter item is stated separately and not as part of the ecosystem due to the way we look at token shadows: in our vision, the software comprising the token shadows is injected onto the core and residential servers either from authorized entities outside the ecosystem or it might be injected to residential servers by the core servers. The token shadow instances are thus the organisms "living" in the virtual ecosystem.

3.4.1 Physical Token Software

In our model, the physical tokens only have minimal functions to allow small embedded devices made of components with very limited hardware resources. Depending on the actual hardware platform, a token could still be able to execute certain processes, which could possibly help (depending on the situation) reducing communication overheads and saving power.

3.4.2 Virtual Ecosystem

The virtual ecosystem is responsible for constructing the respective token shadow for the physical token and make sure that the shadow is able to communicate with the token.

Fig. 3 shows the stack of the virtual ecosystem's components (residential servers in Fig. 2). The bottom two layers are currently covered by the operation system and the Java VM. The server function layer and thus the token shadow layer are implemented in Java. The challenge of the virtual ecosystem is to

- hide the communication details for components of the ecosystem to talk to each other (cf. Fig. 4) and for the communication between the ecosystem and the physical tokens,
- construct the shadows for newly connected tokens and
- update the shadow data for tokens whose state has changed.

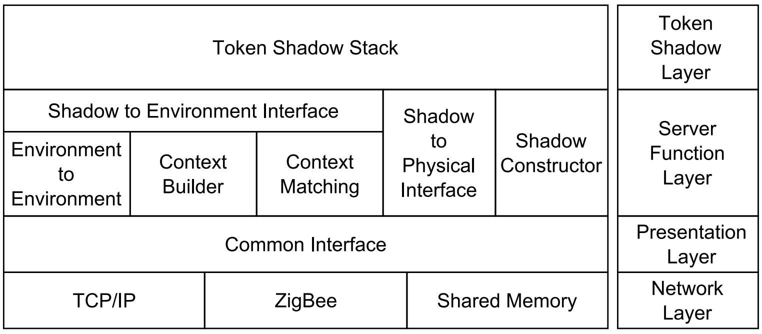

Figure 3: The software stack of the components constructing the virtual ecosystem (residential servers).

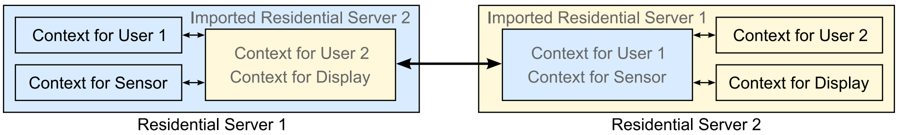

Figure 4: Residential servers talking to each other by importing remote data over the transparent server function layer.

The shadow to shadow communication is abstracted by the server function layer that is responsible for

- handling the context builder, which will register the context of the shadow (what data is supplied for the display tokens, what data is gathered by the sensor tokens),
- matching context information, i.e. forwarding sensed data to shadows that require the respective sensor information or polling sensor data whenever required by another shadow (based on formal context descriptions) and
- handling the communication between the residential servers by querying the core servers for the connection settings for remote residential servers and instantiating the respective communication channels.

3.4.3 Token Shadow Instances

The token shadow instances are the glue between various residential and core servers of the virtual ecosystem. While the residential servers provide the required infrastructure for further information and data exchange, the token shadow instances define what data is provided or required and when and where it is needed (cf. Fig. 5).

The token shadows mainly have two layers, the driver layer, which integrates with the ecosystem and provides a uniform interface to the application layers and the application

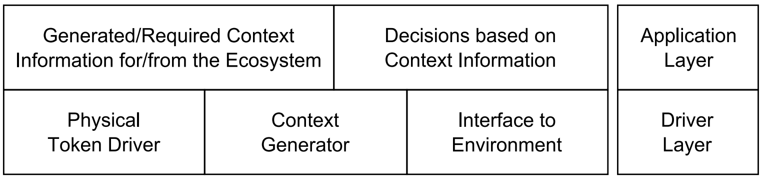

Figure 5: Software stack of the token shadow.

layer, which implement the functionality of the shadow including reasoning, art design or e.g. advertisement related decisions.

4 A ZigBee based Tokenized Interaction Platform

As a proof-of-concept and first demonstration of the platform, we are developing a ZigBee based tokenized interaction platform for implicit interaction in urban areas, especially for shopping mall scenarios, as those are typically crowded public places with the advantages of indoor settings (no heavy rain showers, sun or wind; electricity to run the components required for this platform).

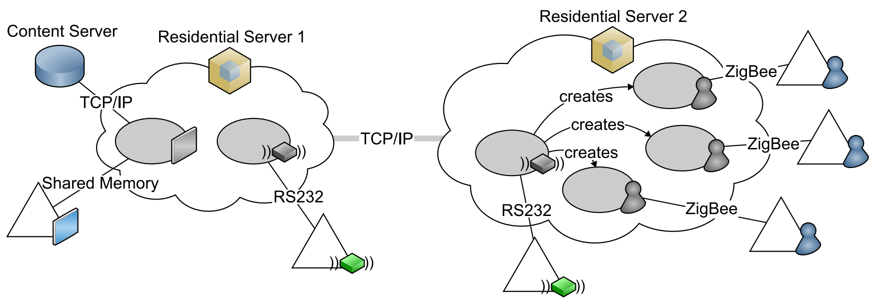

Figure 6: ZigBee based tokenized interaction platform.

The system, as depicted in Fig. 6, is mainly constructed by two residential servers. Server 1 runs two token shadows (one for the sensor token connected via RS232 and one for the display token running natively on the residential server, i.e. the display is connected to the graphics adapter of the residential server). Server 2 incorporates a ZigBee module, connected to the server via RS232, that is able to sense the presence of user tokens over its ZigBee radio interface. It reads out the contextual information stored on the user tokens (user profiles) and instantiates corresponding user token shadow instances on the residential server. The ZigBee communication was implemented using a ZigBee development kit by Texas Instruments, namely the CC2431ZDK. The sensor token on running on server 2 is a TI CC2430 evaluation board, while the user tokens are TI CC2430 demonstration boards, both running our firmware. The sensor token on server 1 is an Arduino Duemilanove which is interfacing ultra sonic sensors to gather environmental information (distances of people nearby). The residential server software stack is implemented in Java SE on a PC platform. While the content to be displayed on the display is planned to be stored in a remote location, it is currently stored on a hard disk inside the residential server 1. The two servers communicate with each other over a socket based communication link using a stream of seralized objects.

The basic idea behind this hardware and software development effort is a shopping mall scenario in which a digital display that is part of a digital signage system can select the content to display by evaluating the current context it is situated in. For this specific scenario this includes information from the ultra sonic sensors (how close are passers-by) and user profile data provided by user tokens. The user profiles depict interests and some basic information about the user and can thus help the digital signage system to display matching content. Of course, it could also just help users find their shops of interest or show the way to the closest money teller or toilet, if required. A person visiting a shopping mall could define his/her personal profile prior to leaving for the shopping mall; once there, the digital displays around would help him/her to navigate through the mall, present special offers and even give discount in the target shops for using the tokenized interaction system. For the shops in the mall this system allows to promote their products and offers target group specific with a probably higher hit rate than a typical mall signage video loop would allow. This, in turn, could lead to a new billing system on the mall signage operator's side: Pay-per-Visualization or Pay-per-Trusted-Contact. The former meaning a payment model linked to the air-time of the displayed content, the latter describing a payment model linked to the number of profile matches found during corresponding content replay.

With the current setup, the content server needs to have its content annotated with meta data to let a context matcher find good matches for a certain user profile. The data gathered from the distance sensors could then be used to choose from a number of different visualization options that have been prepared for various viewing distances. That way a commercial shown on the digital screen could even smoothly transform from a far distance still image to a close distance video visualization while a user would come closer to the display. Please note, that such a functionality is currently not implemented but an option for further development tasks.

5 Conclusion

In this paper we have proposed a tokenized interaction architecture and presented the early stage of a ZigBee based implementation thereof. We are aware of the fact that user profiles and their usage could conflict with privacy concerns, however at this stage of development we didn't want to discuss this in detail but just keep in mind that this could be an issue in real deployment scenarios. However, we think that a user token, following the user on its way through urban areas, could become some sort of a bonus card, totally accepted by users: a user would collect bonus points by passing by displays that can read out the personal profile. When buying a product in a store supporting this tokenized interaction platform, the user would receive a discount e.g. linked to the number of contacts with displays.

We are currently developing both the user token hardware/firmware and the residential server software stack further to have a running prototype with the functions described in this paper within the next few months. Since digital signage products are already out in the field that solve the distribution of content to the screens, the tokenized interaction architecture also provides an interface for applications outside the ecosystem: a content provider could query the system through one of the core servers to get contextual information and use the response for deciding on the kind of content to display. That way, existing digital signage infrastructure could be used to display advertisements with respect to current and maybe even future situations.

6 Acknowledgments

This work is supported under the FFG Research Studios Austria program under grant agreement No. 818652 DISPLAYS (Pervasive Display Systems) and under grant FACT (Flexible Autonomic Context Technologies) of Siemens AG, CT-SE 2, Munich.

References

[BCA+08] V. Barbosa, A. Carreras, H.K. Arachchi, S. Dogan, M.T. Andrade, J. Delgado, and A.M. Kondoz. A scalable platform for context-aware and DRM-enabled adaptation of multimedia content. *Submitted to ICT Mobile and Wireless Communications Summit*, 2008.

[BDO+04] J.F. Bradley, B.R. Duffy, G.M.P. OHare, A.N. Martin, and B. Schön. Virtual personal assistants in a pervasive computing world. In *Proceedings of IEEE Systems, Man and Cybernetics, UK-RI 3rd Workshop on Intelligent Cybernetic Systems, ICS04*, pages 7–8, 2004.

[BFS+08] D. Byrne, J. Freyne, B. Smyth, A.F. Smeaton, and G.J.F. Jones. Classifying public display systems: an input/output channel perspective. In *Workshop Proceedings of Designing and Evaluating Mobile Phone-Based Interaction with Public Displays (co-located with CHI 2008 Conference on Human Factors in Computing Systems)*, April 2008.

[BK01] John Barton and Tim Kindberg. The Cooltown User Experience. Technical Report HPL-2001-22, Internet and Mobile Systems Laboratory, HP Laboratories Palo Alto, February 2001.

[BL97] P. Baudisch and D. Leopold. User-configurable advertising profiles applied to Web page banners. In *Proceedings of the first Berlin Economics Workshop*, pages 24–25, 1997.

[BLSC01] T. Bozios, G. Lekakos, V. Skoularidou, and K. Chorianopoulos. Advanced techniques for personalized advertising in a digital TV environment: The imedia system. In *Proceedings of the eBusiness and eWork Conference*, 2001.

[BPS+08] Timothy W. Bickmore, Laura Pfeifer, Daniel Schulman, Sepalika Perera, Chaamari Senanayake, and Ishraque Nazmi. Public displays of affect: deploying relational agents in public spaces. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3297–3302, New York, NY, USA, 2008. ACM.

[CFJB05] H. Chen, T. Finin, A. Joshi, and MARYLAND UNIV BALTIMORE. *Semantic web in a pervasive context-aware architecture*. Defense Technical Information Center, 2005.

[DAS01] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2):97–166, December 2001.

[Gra02] E. Gratton. M-commerce: The Notion of Consumer Consent in Receiving Location-Based Advertising. *Canadian Journal of Law and Technology*, 1(2):59–77, 2002.

[MAMI07] A.H. Morales-Aranda and O. Mayora-Ibarra. A Context Sensitive Public Display for Adaptive Multi-User Information Visualization. In *Proceedings of the Third International Conference on Autonomic and Autonomous Systems*. IEEE Computer Society Washington, DC, USA, 2007.

[MKHS08] Hema Mahato, Dagmar Kern, Paul Holleis, and Albrecht Schmidt. Implicit personalization of public environments using bluetooth. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3093–3098, New York, NY, USA, 2008. ACM.

[RC02] Anand Ranganathan and Roy H. Campbell. Advertising in a pervasive computing environment. In *WMC '02: Proceedings of the 2nd international workshop on Mobile commerce*, pages 10–14, New York, NY, USA, 2002. ACM.

[RDG+08] Jan S. Rellermeyer, Michael Duller, Ken Gilmer, Damianos Maragkos, Dimitrios Papageorgiou, and Gustavo Alonso. The Software Fabric for the Internet of Things. In C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern, and S.E. Sarma, editors, *Proceedings of the International Conference on the Internet of Things, Zurich, Switzerland*, volume 4952 of *Lecture Notes in Computer Science*, March 2008.

[SBHM02] P. Sutton, M. Brereton, C. Heyer, and I. MacColl. Ambient Interaction Framework–Software Infrastructure for the Rapid Development of Pervasive Computing Environments. In *Inaugural Asia Pacific Forum on Pervasive Computing*, page 1. Australian Computer Society, 2002.

[SK05] Christian Sandor and Gudrun Klinker. A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality. *Personal Ubiquitous Computing*, 9(3):169–185, 2005.

[SPD06] Matthew Sharifi, Terry Payne, and Esther David. Public Display Advertising Based on Bluetooth Device Presence. In *Mobile Interaction with the Real World (MIRW 2006) in conjunction with the 8th International Conference on Human Computer Interaction with Mobile Devices and Services*, September 2006.

[SRA09] Stabeler M. Shannon R. and Quigley A. Profiling and targeting opportunities in pervasive advertising. In *Pervasive 2009 workshop on Pervasive Advertising*, Nara, Japan, 2009.

[VSdI06] J.I. Vazquez, I. Sedano, and D.L. de Ipina. Evaluation of orchestrated reactivity of smart objects in pervasive semantic web scenarios. In *Proceedings of The Second International Workshop on Semantic Web Technology For Ubiquitous and Mobile Applications (SWUMA06) at the 17th European Conference of Artificial Inteligence (ECAI 2006)*, 2006.