En passant Coupon Collection

A. Ferscha, W. Swoboda and Ch. Wimberger
Johannes Kepler University Linz, Department of Pervasive Computing
Altenberger Straße 69, 4040 Linz, Austria
{ferscha, swoboda, wimberger}@pervasive.jku.at

Abstract: Spontaneous interaction in public places has evolved as crucial concern in interaction design, particularly in the domain of public advertising. Implicit interaction is a mode of spontaneous interaction in which the user is not attentively or explicitly expressing input to a system, but the system from observing the user, and considering all the available information that describes the user situation, is used to generate input - and behaves accordingly. We have investigated implicit interaction as a potentially effective means to design for spontaneous interaction in the public. An *en passant* coupon collection system, CCS, has been developed to implement an implicit interaction based pervasive advertisement process. CCS is based on scanning and identifying Wi-Fi hotspots by their BSSID, and has been implemented for internet enabled smart phones. Scenarios demonstrate how the system collects bonus points, rewards, sales coupons, reward coins, etc. by users just walking by points of interest like stores, bus stops, schools or offices. CCS neither not being reliant to a preconfiguration of the Wi-Fi infrastructure, nor its use for wireless data communication, encourages for a parasitic use of existing Wi-Fi networks (already available in public places like shopping malls, airports, train stops, stations or city centers) as a pervasive advertisement system.

1 Implicit Interaction and Public Displays

Pervasive Computing has developed a vision where the “computer” is no longer associated with the concept of a single device or a network of devices, but rather the entirety of situative services originating in a digital world, which are perceived through the physical world. It is expected that services with explicit user input and output will be replaced by a computing landscape sensing the physical world via a huge variety of sensors, and controlling it via a plethora of actuators. The nature and appearance of these services will change to be hidden *”in the fabric of everyday life”*, invisibly networked, and omnipresent. They will greatly be based on the notions of context and knowledge, and will have to cope with highly dynamic environments and changing resources.

Interaction with such computing landscapes will presumably be more implicit, at the periphery of human attention, rather than explicit, i.e. at the focus of attention: *”we will be able to create (mobile) devices that can see, hear and feel. Based on their perception, these devices will be able to act and react according to the situational context in which they are used”* [Sch00]. Accordingly, as from the users point of view, *”implicit human computer interaction is an action performed by the user that is not primarily aimed to in-*

teract with a computerized system but which such a system understands as input." [Sch00]. Implicit interaction is hence based on two main concepts: perception and interpretation. Perception is information gathering about the environment and situations, usually involving (technological) sensors. This information is generally provided implicitly to the system and displayed naturally to the user. Interpretation is the mechanism to understand the sensed data. Conceptually, perception and interpretation when combined are described as situational context. Clearly, *pervasive advertisement*, like e.g. advertisement via *public displays*, is a case demanding for principles of implicit interaction.

Public display systems (PDS) [FV02], often referred to as *Out-of-Home Media Systems*, by gaining from technological advances in display technology, but also wireless communication technologies, have emerged as such a landscape, provisioning and controlling content related to *context*. "Context" refers to any information describing the situation of an entity, like a person, a thing or a place, so the content being displayed in a PDS (in e.g. shopping malls, bus and train stations, airports, offices, schools, public places, etc.) can be adapted wrt. to all these types of information.

While user interaction with PDS was not intended over their first generations of existence, the upcoming electronically operated and digitally controlled DSPs have encouraged a variety of solutions for *explicit interaction* [SD08], [BRS05], [BBRS06], [VB04] mostly being based on mobile phones as the interaction device [HR08], [WFGH99], [TCF$^+$07], the primary types of interaction being navigation and browsing [RYL$^+$07] [WK06].

Interaction with such PDS landscapes will presumably be more implicit, at the periphery of human attention, rather than explicit, i.e. at the focus of attention.

With this paper therefore we address *implicit interaction* with PDSs, addressing the issues of a seamless, non-disrupting, yet situation aware modes of interaction in public advertisement. We have developed a Coupon Collection System (CCS) based on (even non-cooperation) scanning for public Wi-Fi hotspots, and the mapping of unique hotspot IDs to geo-referenced advertisement services. The Coupon Collection System is developed in Section 2, and the roles of stakeholders engaging in a pervasive advertisement service system are related in Section 3. The corresponding software architecture and implementation details are outlined in Section 4. Section 5 exemplifies the use of the CCS system with smart phones as context aware CCS clients. Conclusions are drawn in Section 6.

2 The CCS Coupon Collection System

CCS is divided into three major components. There are stationary components, which are installed on places where a user should be able to collect coins. The users carry the mobile components, which detect the stationary components automatically, without any explicit interaction. Finally, a server is needed, to manage the content which will be delivered to the mobile components.

2.1 Stationary Component - WI-FI Access Points

The major task of a stationary component is to be detected by the mobile component in a specific radius around its geographical location. One option to do that, without provoking an explicit interaction by the user, is to broadcast a radio signal with limited range Fig. 1. Of course, if there are two or more stationary components with overlapping radius, the interference between that signals has to be considered. To distinguish between different stationary components, each stationary component also needs a unique identifier.

There is an existing and also very common technology which meets the requirements, the 802.11 wireless LAN access point. A functionality provided by access points is the beacon broadcast - also known as SSID-Broadcast-, a specific package, sent at certain intervals. Usually the beacon broadcast is meant to make the access point known to wireless client devices in vicinity, therefore the beacon packages are not encrypted. Such a package contains, among other information, the MAC address of the access point, the name of the network, the time interval between the broadcasts and the encryption of the network [IEE07]. Using the beacon package, a client device is also able to evaluate the signal strength. Important for this system is mainly the MAC address of the access point for the use as a unique identifier. Usually, the interval between the beacons is set to 100 milliseconds. That leads to a very fast possible reaction time for the client. Other technologies are much slower, for example Bluetooth (with query scan interval set to default, the reaction can take up to 2.56 seconds [SIG09]). Also the frequency bands which are used by 802.11 Wi-Fi networks (2,4 and 5,18-5,7 GHz) are usable without a license.

Figure 1: access points with SNR visibility: (from left to right) (i) one AP, (ii) multiple APs, (iii) multiple APs with overlapping SNR visibility, (iv) Possible Distribution of Wi-Fi Access Points in public space

The interference management is provided by the 802.11 standard and supported by the hardware of access points [Ohr03]. That fact is important for the scalability of the system, because thereby overlapping domains are no longer a problem. The effective range of wireless LAN depends on factors as the transmission power, the used antenna and of course the consistency of the surroundings. As a reference value, 802.11b standard access points can reach a radius of up to 100 meters. To limit the radius of the access points it's either possible to decrease the transmission power, which is does not work on every access point, or to filter signals under specific signal strength on the side of the mobile component [Ohr03]. If the signal strength of an access point is under certain value, the mobile component will just ignore it in first place or doesn't redeem an BSSID for a coin when connected to the server. This feature doesn't exist in the prototype by now, but it's a likely target for further development. As an disadvantage, specification of a collection area for a specific virtual coin is not very accurate. It's not possible to define precise boundaries which separate one collecting area from a non-collecting area or an area of another coin. For operating the CCS one has to be aware of this disadvantage and compensate it by clever distribution of access points and mapping of coins.

In areas where there is already a wireless LAN infrastructure , the operator can call on existing access points. In fact, every access point with enabled beacon broadcast can be used to act as a stationary component in this system, even if it doesn't belong to the operator of the system.

2.2 Mobile Component - Smart Phone

The requirements a mobile component has to meet to work in this system are matched by many of the current and future generation smart phones. They provide the required connectivity like wireless LAN for collecting BSSIDs and UMTS for connection to a server on the web, more than enough computing power and storage, as well as high resolution displays and keys or even touch-screens for interaction with the software. However, not every smart phone is equally well qualified for the use in this system, even if it meets all hardware requirements.

The mobile component of the system is implemented as software on a smartphone. If the user wants to start collecting, he has to activate the software. From that point of time, until he switches it off again, the software collects coins.

An important point which has to be considered is the increased power consumption. Without a program running, a smart phones switch into an energy saving mode, where all but the most basic functions are deactivated. While the software searches for access points, the phone will not be able to do that, which leads to a significantly reduced operation time.

The prototype for the mobile component of the system was developed for the Apple iPhone, a quite innovative and widespread piece of technology.

2.3 Server

After the mobile component collected for a while, its database has stored some MAC addresses. It's not known yet what MAC address belongs to what coin. So at certain points in the program the mobile phone has to connect to a database, where a mapping between MAC addresses and coins happens. Keeping this database on the phone itself leads to various problems. How to keep the database up to date? What can be done to keep it save from malicious manipulation? In addition there might be a lack of storage space on the device. Therefore the database is not located on the phone itself but on a web-server. As a result the mobile device needs internet connectivity, which should be no problem in times of UMTS and wireless LAN hotspots. As the prototype exists now, it's not possible to regard the time when a token was collected. But it's an idea for future versions to store timestamps with collected BSSIDs and make the type of collected coin not only dependent on the BSSID but also on the time when it was collected.

3 Stakeholders and the CCS Service Architecture

For a pervasive advertising scenario, the individual stakeholders engaging in CCS services and their roles are sketched in Figure 2(left). The first stakeholder is the operator of the system. It's the one who sets the system up and maintains it. The operator also contracts with the businesses the system advertises for. Those businesses are the second stakeholder. The third party are the customers of the businesses, the ones the advertisement is addressed to. This third stakeholder is in possession of the platforms for the mobile components of the system, the smart phones. The operator owns the right on the software. The property situation between operator and businesses - who owns the access points and who runs the server - can be different from scenario to scenario.

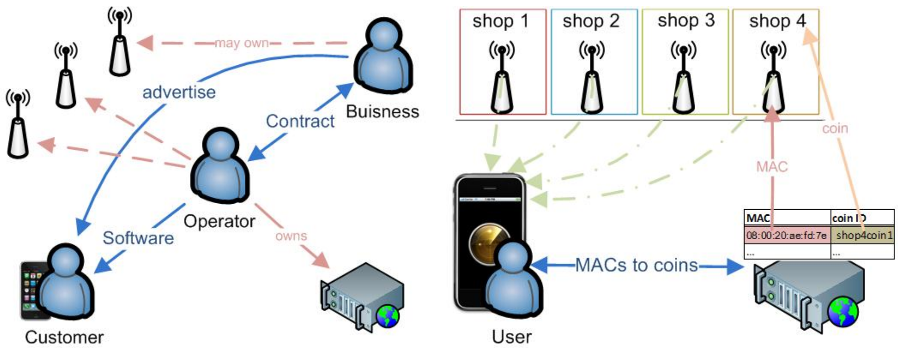

Figure 2: left: Role and Interrelationship of Stakeholders; right: System Architecture and Mapping

3.1 Setting up a CC System

A good way to describe the architecture of this system is to go through a setup process. Therefore a short scenario has to be explained.

In this scenario there is a section of a street with a number of shops or other businesses. Each of this business wants to participate in the system. For that reason they enter into a contract with the operator of the system. The later user of the system, the customer, should earn coins by walking down this street. When he gets near a shop he gets a coin of this shop.

For that to work, the operator of the system has to install one wireless access point, one of the stationary components, inside or near each shop. He doesn't have to add this access points to a network, he just has to switch them on and configure them. However, the operator might want to provide additional services like internet connectivity over those access points, either for the shops or for customers. Doing that wouldn't interfere with the functionality of the system described here. As mentioned before he can also use existing access points, for example if the shopping street already possesses a wireless LAN infrastructure (Fig. 2 right).

After the access points are set up and running they have to be mapped to coins. That happens by storing their MAC address and a reference to a coin to the database on the server. Storing additional information about the access points, like the GPS coordinates, the model or the range, might also prove valuable. The operator can also map one MAC address to more coins and vice versa. By that the user can get some different coins out of one single access point.

The only thing missing now are the mobile components. It's the operator's task to distribute the software, which converts the user's smart phones into mobile components of the system. After the software is installed, the user just has to run it to collect coins. The system is set up and running at this point. It can be easily scaled by adding new mobile and stationary components.

3.2 Interaction between Components

The interaction between the components is quite simple. The software on the smart phones just scans for the beacon broadcasts and stores the MAC Addresses of the access points. As soon as the user wants to check which and how many coins he collected so far, the software connects to the server and gets the proper coin for each MAC address. If the user collected enough coins of a kind to redeem them, for example at a shop, the software connects to the server again to get the information needed to cash the benefit. That could be, for instance, a picture of a bar code [Luf].

4 The CCS Software Architecture

In order to achieve the best possible user acceptance, a clear and straight-forward user interface has to be presented to the person using the software on the smart phone [STH07]. In addition, the software has to be easily operated on the go, because the handling of the coupons after collecting them should be at least as simple as it is with traditional coupons - even for less tech-savvy people. In order for this, we first introduced the metaphor of a "coin" to represent a coupon. A coin in its appearance should offer all the used modalities of handling it, like keeping, saving, or droping it. to manage coins in the user interface of a smart phone, the direct manipulation of visualizations of coins appeared inevitably necessary. Therefore, the Apple iPhone was chosen to purposely serve as the platform for the prototype client. The programming language used is Objective-C as it is the main language for developing on the iPhone platform.

For optimal scalability, even with thousands of collected records, SQLite is used for storage on the client. SQLite is a lightweight and server-less library that implements a transactional SQL database engine. After installing the client software, no additional setup is required to start collecting coins. The entire collecting and redeeming process is completely anonymous.

The server software was developed in PHP with a MySQL database backend. It basically understands three different commands from the client and responds accordingly after issuing some SQL commands to the database. A web frontend can be easily developed to allow third parties to add their coins and access point addresses.

4.1 Scanning for Wi-Fi 802.11 Beacon Broadcasts

Immediately after touching the collector icon on the home screen, the client starts and finds itself in a discovery mode listening for Wi-Fi 802.11 beacon broadcasts. All that is visible to the user at that time is a nearly empty display with an icon displaying a radar screen in the center. In order to keep the system utilization as low as possible, animations are disabled during collecting. The act of finding nearby access points happens in a completely passive way (without sending any Wi-Fi 802.11 packets) thus battery life can be kept at a maximum.

The client continuously keeps an updated list of available access points at the current location. As soon as a new access point appears, it is added to the list and the MAC-address, along with a timestamp, is stored into the database. When an access point in the list cannot be reached anymore, it is removed from the list.

At this point the client doesn't know if the collected MAC-addresses will be exchanged to collected coins in the future. In reality most of the collected addresses will be ignored by the server and only a few will transform into coins. Hence the list of found addresses can potentially be very long, especially if the user keeps collecting over a long period of time without checking for collected coins.

The process of collecting can be interrupted by the user at any time by closing the application and can be continued by simply restarting the application again. By touching the radar screen with his fingers, the user can check if any coins have been collected for him. At this point an internet connection is established and the server is contacted.

4.2 Client-Server Communication

Only at a few certain moments, the server has to be contacted. On one hand when collected MAC-addresses should be exchanged for coins and on the other hand when these coins should be exchanged for the final coupon. To assure the best possible interoperability, the entire communication is based on the standardized Hypertext Transfer Protocol (HTTP). So even in environments with firewalls and proxy servers there should be no problem communicating with the server through the internet. If no internet connected Wi-Fi 802.11 hotspot is around, the iPhone connects to the internet through your mobile provider.

When communicating, the connection is always established by the client and then a HTTP-GET query is issued. Depending on the query, the server responds with a plain text answer or with an image. This answer is then parsed by the client.

There are basically three commands which can be sent by the client.

- getCoins: The client sends his device id and a list of BSSID and timestamp pairs. The server responds with a plain text answer that is composed of multiple text lines. Each line can be of one of three types. A coininfo record containing meta information regarding the collected coin, a bonus record describing which bonus is available for a type of coin and a coin record itself containing a server generated coin id and the type of the collected coin. When a coin type is received for the first time, the coin image has to be downloaded from the server with the following command.
- getCoinImage: The client sends the id of the coin type for which the image is needed and the server directly responds with the image data encoded in the PNG file format. This image has a 80 x 80 pixels dimension.
- getBonus: The client sends his device id, the id of the coin type it is trying to redeem, the amount of coins it is sending and a list of previously collected coin ids. The server responds with the image data of the bonus encoded in the JPEG format. This image has a 320 x 480 pixels dimension to fit exactly the screen of the device. This image can be dynamically generated on the server and a barcode can be included for easier redeeming.

Example getCoins query:

```
http://collector.server.com/collector/api.php?action=
getCoins&did=B1D35D01-32DA-590B-9518-BFDB8F5BC4B8&
data=0:1b:63:2b:4a:b7-1243326354;
```

Answer from the server (fields in the lines are tab separated):

```
I euNCXjTOOJ Plus City http://www.plus-city.at/
B euNCXjTOOJ 10 5 Euro Gutschein
C dn2ccssrXp euNCXjTOOJ
```

```
I f1GRgsU3z7 Müller http://www.mueller-drogerie.at/
B f1GRgsU3z7 10 -10 % auf alles
C d66nrs5cyT f1GRgsU3z7
I skb2a95j2G Cosmos http://www.cosmos.at/
B skb2a95j2G 10 -5 % auf alles
C 9QIKl7O5DD skb2a95j2G
```

4.3 The Coin Management Interface

After touching the radar icon in the collecting screen a new screen for managing the collected coins appears. After the server responded to the getCoins command issued by the client, all newly collected coins appear at the top border of the screen. These coins are branded with the store's logo and can be dragged with the fingers to two drop zones at the lower border of the screen. The left drop zone represents the already collected stack of coins. By dropping a coin onto it the coin is added to the stack and is also collected in the future. The drop zone on the right side symbolizes a horizontal slot to throw newly collected coins away. By throwing coins away they are added to an ignore list and won't be collected anymore. So the client learns which coins are of interest for the user and which ones are not.

For easier decision making, the user can touch any coin and is directed to a page with additional information about that coin, the issuer and the available bonuses for this kind of coin. The additional information can be plain text, a web page or even a video describing the benefits of collecting this coin.

By touching the throw away slot a list of ignored coins is displayed. A coin can be removed from the ignore list as well, but previously ignored coins have to be recollected again - they are not saved.

To get a detail view on the collected coins the user simply needs to touch the stack of coins on the bottom left corner. He then gets to a detail screen where the collected coins are displayed together with the available bonuses. Each type of coin has its own page and by swiping over the screen the user can switch between these pages. With this feature the user is able to get a quick overview of collected coins and how many still have to be collected to receive a coupon. Of course the user can also go to the coin description from this page.

4.4 Screenshots of the User Interface

From left to right Fig. 3 shows the client in collecting mode, the coin organizing screen, the detail collection view and the final coupon.

Figure 3: Screenshots of the collecting and redeeming chain

4.5 Security/Abuse Considerations

To not let the user show his coupon to more than one cashier, the screen displaying it is discarded after a limited time. In addition, a little animating element as pictured in the rightmost screen in Fig. 3 stops the user from taking screenshots from his screen and using this screenshot afterwards.

To ensure that nobody tries to ask the server for a bonus with faked coins, every virtual coin issued by the server has a globally unique, random id. This id is also stored on the server and compared to the ids coming from the client. Even in the case of smartphone backups where parts of it's file system can be restored to a previous state, the coins are marked as redeemed on the server and cannot be redeemed again. To ensure that no fake requests for a high amount of coins are accepted, the server limits the amount of coins issued per client per access point in a certain time span. A scenario with a user living near an access point and collecting the same coins every day is not considered as fraud. The user simply doesn't get a bonus for every coin. Patterns of misuse can be filtered on the server and bad clients could easily be banned for some time.

5 Scenarios

While the CCS was developed for a very wide scope of applications, the following scenarios are among the most obvious ones.

5.1 En passant Coupon Collection while Shopping

The user's activity in this scenario is shopping. It can take place in a mall, a shopping street, or even a whole city. The dimension of the system in this scenario is very well scalable.

The operator of the system has contracts with the businesses in the shopping areas. That can be, for example shops, cafes, restaurants or kiosks. One thing those businesses want is advertisement. Another benefit they might get is information about the customers. Information about the routes they take through a shopping area, how fast they walk through, at which shops they stay for a longer time or which offers they are interested in might prove valueable. The operator is responsible for maintaining the system, distributing the software needed for die client devices, and promotion of the system.

Of course, there has to be some benefit for the customers as well or they would never take action to get and install the software on their smart phones. Therefore they get bonuses from collecting the virtual coins. In this scenario the bonus would be, for instance, a coupon to get discount at a certain store.

So the operator installs access points at various locations, at advertisement walls, entrances and of course the shops. He maps the access point to the coins. For instance, in exchange for passing an advertising wall where Shops A, B and C are advertised, the customer should get coins for shops A, B and C. When passing shop D, he should get three coins for shop D. To make that work the operator maps the MAC of the one access point near the advertising wall to shop A-, B- and C-coins, the access point inside shop inside shop D with three shop D-coins. Some of the bigger stores have their own coins, some of the smaller ones, the cafes in the area for example, have one joint coin together, so that the customer can choose in which one he wants to redeem his coupon.

So, the customer walks through the shopping area. He gets to a shop, shop A (Fig. 4), but he is not very interested in what shop A has to offer, so he walks by without entering. Nevertheless the software on his phone collects the shop A coin. He continues ambling through the area, peeks in shop windows, enters some of the shops to take a look at the offers. By that he collects coins for many of the stores around, also some more shop A-coins.

Figure 4: collecting a coin

Figure 5: redeeming a coupon

Now, aware of the fact that he should have collected some coins by now, he takes a look at his smart phone. He browses through the coins and looks up what bonus or coupon he gets for what type of coin and what quantity he needs to redeem them. He deletes all coins he is not interested in, and keeps the ones which he intends to use in future. The coupon he gets from collecting shop A-coins attracts him and he has already collected enough shop-A coins to redeem it, so he goes back to shop A to buy the article. Now, at the cashier, he wants to redeem the minus twenty percent discount he gets on the article he wants to buy. Therefore he switches to the redeeming screen on his phone and shows it to the cashier. The redeeming screen can show, for instance, a barcode or a simple number code Figure5 [Luf]. After he redeemed the coins, the coins are gone and can't be redeemed again until he collected new ones.

After a long day of shopping, both, the customer and the businesses are satisfied. The customer, because he got some things below the ordinary retail price. The businesses achieved that the customer stood longer in the shopping area than he has intended initially and maybe bought some things he didn't intend to buy in first place. In addition, the businesses can analyze the customer's behavior by taking a look at the order in which the coins were collected, the time between collection of coins, the time the customer stood inside the area and so on. This data might help improving the attractiveness and therefore the profit of the shopping area.

5.2 Coupon Collection while Traveling

Yet another scenario is traveling. There are several possibilities where the system can be used by a traveler. It mostly depends on how he travels, be it by car, by public transportation or by plane.

This scenario might be a bit more difficult for the operator. He is responsible to distribute the access points. In the shopping scenario, even if the shopping area is big, it's manageable. For the traveling scenario we need access points almost everywhere, at train stations, airports, gas stations, advertisements beside roads and in the ideal case everywhere along every root a traveler can take. This, of course, is hardly possible, but there is a solution for an approximation to this. As mentioned before, the operator doesn't have to own the access points used by the system. He can use existing access points. That can be public access points, like the internet hotspots in cities, private access points which are placed in apartments or access points for the internal network of companies. The MAC addresses have to be mapped to coins, which is a bit of an effort, but much cheaper than setting up own access points.

Naturally using foreign access points has disadvantages as well. The reliability is lower and the maintaining costs are higher, because private access points might be removed or exchanged. Some of them are highly reliable, though. Public hotspots, for instance. Anyway, on places where reliability is crucial the operator might want to install his own access points.

Now let's assume the traveler travels by car. He has to drive a long distance between two cities. He travels very frequently and is aware of the benefits the usage of the system can provide. So, at the beginning of the trip, he activates the software on the phone. After a while of driving he needs to refuel. Now he could pick any gas station, but, while driving, he might have collected a number of coins of a certain gas company. That can be, for example, at advertisements beside the road or gas stations of the same company. So he checks his smart phone and realizes that he has enough coins to redeem a minus ten percent coupon at gas company B. He makes his choice of a gas station conditional on the amount and type of coins he collected.

Sadly, after arriving in the destiny city his car breaks down beyond repair. So, in order to get home after he finished his business in the city, he uses public transports. He is used to go by public transports and is also aware of the advantages the coupon collecting system can provide in this situation. A bus company has contracted with the systems operator. In previous visits he has collected some of the company's coins. Now he is able to redeem a coupon to get a short distance with the bus for free.

Later he arrived at the train station. He is a frequent train user and has some train coins on his stack. He collected them by doing some shopping in the train station, passing by train company advertisements and of course by using the train. So he gets his train ticket cheaper by redeeming the train company coupon. If he had chosen to go by plane, some similar benefit would have been possible, like some kind of frequent flyer's miles.

The benefit for the contractors with the system's operator is at the one hand direction of the customer. With clever usage of the system they can make a customer to choose their business over other businesses. On the other hand they can reinforce the relationship to existing customers by giving them benefits for frequent usage of their offerings. Again, it's possible to analyze the data the system extracts out of the user behavior.

6 Conclusions

Pervasive advertisement systems in the future will be crucially reliant to modes of spontaneous, implicit interaction with users: implicit interaction is a mode of spontaneous interaction in which the user is not attentive to a system or the environment. Also, the user is not, an cannot be expected to give explicit response to an interaction system in public space. After investigating on implicit interaction as a potentially effective means to design for spontaneous interaction in the public, we have developed an *en passant* coupon collection system, CCS, to support implicit interaction based pervasive advertisement processes on a wide scale of applications. CCS is based on a scanning for Wi-Fi 802.11 beacon broadcasts, and has been implemented for internet enabled smart phones, here for the iPhone. With scenarios we can demonstrate how the system collects bonus points, rewards, sales coupons, reward coins, etc. by users just walking by points of interest like stores, train- or bus stops, or just in public places of towns or cities. CCS is neither reliant to a preconfiguration of the Wi-Fi infrastructure, nor does it use for wireless data communication. By that, CCS is a very universal approach of implementing geo-referenced, context-aware, personalized services in pervasive advertisement.

7 Acknowledgements

This work is supported under the FFG Research Studios Austria programme 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

[BBRS06] Rafael Ballagas, Jan Borchers, Michael Rohs, and Jennifer G. Sheridan. The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing*, 5(1):70, 2006.

[BRS05] Rafael Ballagas, Michael Rohs, and Jennifer G. Sheridan. Sweep and point and shoot: phonecam-based interactions for large public displays. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1200–1203, New York, NY, USA, 2005. ACM.

[FV02] Alois Ferscha and Simon Vogl. The WebWall. In *Proceedings of the Ubicomp 2002 Workshop on Collaboration with Interactive Walls and Tables*, Göteborg, Sweden, September 2002.

[HR08] Robert Hardy and Enrico Rukzio. Touch & interact: touch-based interaction of mobile phones with displays. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 245–254, New York, NY, USA, 2008. ACM.

[IEE07] IEEE. *IEEE Standard paper 802.11 - 2007*, 2007.

[Luf] Lufthansa. Lufthansa boarding passes using barcode on mobile phone displays.

[Ohr03] Roeder Ohrtman. *WI-FI Handbook 2003*. McGraw-Hill, 2003.

[RYL+07] Han-Sol Ryu, Yeo-Jin Yoon, Myeong-Eun Lim, Chan-Yong Park, Soo-Jun Park, and Soo-Mi Choi. Picture navigation using an ambient display and implicit interactions. In *OZCHI '07: Proceedings of the 19th Australasian conference on Computer-Human Interaction*, pages 223–226, New York, NY, USA, 2007. ACM.

[Sch00] Albrecht Schmidt. Implicit human computer interaction through context. *Personal and Ubiquitous Computing*, 4(2):191–199, June 2000.

[SD08] Corina Sas and Alan Dix. Designing and evaluating mobile phone-based interaction with public displays. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3941–3944, New York, NY, USA, 2008. ACM.

[SIG09] Bluetooth SIG. *Bluetooth core specification v3.0*, 2009.

[STH07] Albrecht Schmidt, Lucia Terrenghi, and Paul Holleis. Methods and guidelines for the design and development of domestic ubiquitous computing applications. *Pervasive Mob. Comput.*, 3(6):721–738, 2007.

[TCF+07] Nick Taylor, Keith Cheverst, Dan Fitton, Nicholas J. P. Race, Mark Rouncefield, and Connor Graham. Probing communities: study of a village photo display. In *OZCHI '07: Proceedings of the 19th Australasian conference on Computer-Human Interaction*, pages 17–24, New York, NY, USA, 2007. ACM.

[VB04] Daniel Vogel and Ravin Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 137–146, New York, NY, USA, 2004. ACM.

[WFGH99] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. Bridging physical and virtual worlds with electronic tags. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 370–377, New York, NY, USA, 1999. ACM.

[WK06] Hendrik Witt and Holger Kenn. Towards implicit interaction by using wearable interaction device sensors for more than one task. In *Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, page 20, New York, NY, USA, 2006. ACM.