Bringing Semantic Services to Real World Objects

Massimo Paolucci[a], Gregor Broll[b], John Hamard[a],

Enrico Rukzio[c], Matthias Wagner[a], Albrecht Schmidt[d]

[a] DoCoMo Communications Laboratories Europe GmbH
Munich, Germany
{lastName}@docomolab-euro.com
[b] Medieninformatik, Ludwig-Maximilians-Universität
Munich, Germany
gregor.broll@ifi.lmu.de
[c] Computing Department, Lancaster University
Lancaster, UK
rukzio@comp.lancs.ac.uk
[d] Institute for Computer Science and Business Information Systems
University Duisburg-Essen
Essen, Germany
albrecht.schmidt@ifi.lmu.de

Abstract. The last few years have seen the emergence of two parallel trends: the first of such trends is set by technologies such as Near Field Communication, 2D Bar codes and RFID that support the association of digital information with virtually every object. Using these technologies ordinary objects such as coffee mugs or advertisement posters provide digital information that can be easily processed. The second trend is set by (semantic) Web services that provide a way to invoke automatically functionalities across the Internet lowering interoperability barriers. The PERCI system, discussed in the paper, provides a way to bridge between these two technologies allowing the invocation of Web services using the information gathered from the tags effectively transforming every object in a service proxy.

KEYWORDS

Human-Machine Systems, Web Technologies, Mobile Technologies, Semantic Data Model

INTRODUCTION

In recent years, tagging technologies, such as Near Field Communication (hereafter NFC) (ECMA 2006), Radio Frequency Identification (RFID) (EPC Global 2006) and 2-dimensional bar codes (Adams, 2007) have received a great deal of attention because of their ability to associate digital information to arbitrary objects. In principle everything can be cheaply tagged, from coffee mugs to advertisement posters, and applications can be invoked on the basis of the information that is gathered from the tags on the object.

Japan has seen initial commercial deployments of these technologies. There 2D tagging of advertisements, an example of which is shown by the banner in Figure 1, is quite common. By taking a picture of the banner, a mobile user would immediately gain access to an airline Web page and there buy tickets to fly anywhere in the World. NFC tagging in combination with mobile phones is also widely used in Japan through iMode-FeliCa (Yoshinaga, 2003). Tagging technologies are also becoming increasingly common in Europe and the US. For example, the UK company Glue4 is delivering platform to associate NFC tags to advertisement posters (UsingRFID.com, 2007), while the German railroad operator uses 2D bar codes to store train tickets directly on the phone (Deutsche Bahn 2007). Furthermore, tagging is widely used also in B2B solutions to monitor stock inventory, logistics and transport to track moving goods (Meloan, 2003). By tagging containers as well as single products, companies can easily manage their inventory, keep track of the goods that have been received and monitor their position during the delivery process (IDTechEx, 2007)(Tohamy, 2005).

Whereas tagging is entering the computer science main stream, it also provides new challenges. The first one is that tags, by and large, do not have any processing abilities, rather they passively return always the same value. Even when active tags exists, as in the case of NFC and RFID, their cost is approximately 100 times higher of the corresponding passive tags making the broad use of active tags very costly. The second challenge is that tags have very limited memory storage, which ranges from few bytes to few Kbytes. The third challenge is that the range of the "network" to read a tag is also very limited, spanning from a few centimeters in the case of NFC tags, to few meters in the case of RFID tags. The fourth challenge, which is important for this paper, is that tags break the traditional structure of SOA-based applications which assumes that services are always available and discoverable through a registry.

Figure 1: A banner with a 2D bar Code in Tokyo

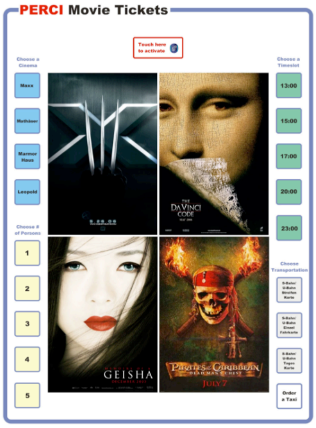

Figure 2: PERCI movie poster

In this paper, we attempt to tackle these challenges by providing a general Web service infrastructure to support the user to interact with advertisement posters. Through the paper, we will use the movie poster displayed in Figure 2 as our main example; the goal of the poster is to advertise movies that are playing in town, but the poster achieves more than that, the little squares on the side of the poster specify additional information such as: the movie theaters where the movies are played (on the upper left side), the show times (on the upper left), and the number of tickets that the user may be able to buy (on the bottom right), public transport tickets to go to the theater (on the bottom left).). Crucially, each one of the little boxes is associated with an NFC tag that can be read from any mobile phone with an appropriate reader. A mobile user swiping her phone on the tags should be able to purchase a ticket to a showing of the desired movie, and a public transport ticket to go to the movie. Ultimately, the poster becomes the user interface of the movie theater ticket selling service.

To exploit tagged objects such as the poster displayed in Figure 2 we introduce the PERCI system (Siorpaes et al. 2006, Siorpaes 2006). PERCI (PERvasive serviCe Interaction) provides an infrastructure that supports users to perform complex interactions with arbitrary objects. The basic idea behind PERCI is to use Semantic Web service technology to associate the object to the corresponding service. Using Semantic Web services, PERCI establishes on the fly a service client on the bases of the information read from the tags on the object, and in the process it configures a user interface to expose the service to the mobile user.

In the rest of the paper, we will discuss the ideas behind PERCI in more detail. In Section 2 we will review existing literature and applications using tags on objects; in Section 3, we discuss the PERCI system and implementation; in Section 4, we will discuss the properties and relevance of the system, and finally in Section 5 we conclude. Throughout the paper we will assume familiarity with the basic Web services standards, such as WSDL (Chinnici, 2006), and with the main ideas of OWL-S (Martin et al., 2007).

RELATED WORK

There are multiple ways to connect services and tags. In the first way, the tag consists of an active reader that is connected to the service, while the user holds a card or a phone with a card embedded in it. To trigger the service, the user swipes a card in front of the reader which reads known location on the card and then communicates the data directly to a service. Eventually, information may be sent back from the service and written on the card as a response. This solution is used in a number of applications from public transport to building access, as well as in location aware systems (Hazas, 2004). But the problem is that this approach is also very rigid when used for service provisioning. Since cards are associated with services, in the worst case, the user needs a different card for each service she uses. Even when the card is shared across applications, as in the case of FeliCa cards on Japanese phones (Yoshinaga, 2003), the memory on the card should be managed carefully to prevent clobbering information across services.

An alternative common way to associate (passive) tags to services is by linking them to web pages. An example of such tags is shown in Figure 1 where a 2D bar code is used to encode the URL of an airline. The mobile phone decodes the image to extract the URL and it displays the corresponding Web page on the browser. The association of tags with URLs has been widely proposed in the academic literature. Most prominently, the Cooltown project (Kindberg and Barton, 2001)(Kindberg et al, 2002) advocated the use of web links to connect tags in the environment to information about different objects. The problem with this approach is that it does not easily support the type of interaction that is required by the poster displayed in Figure 2, because the page to be displayed with such poster depends on the reading of a number of different tags, rather than one single reading. The second problem of this approach is associated with the lack of semantics of HTML. Specifically, while the user can buy the movie ticket from an HTML page, the ticket should be stored on the service site, as it is done for all e-commerce applications running on the Web, instead of being stored on the phone. Therefore the ticket could not be used to relate different services and applications; for instance, it could not be used to tell the user's calendar that that a given time is blocked for the movie.

A number of projects propose a direct access to the service assuming that service clients have been delivered to the phone before the tag has been read. These solutions include iMode-FeliCa that requires an i-appli stub to access any service (Yoshinaga, 2003). A similar approach is adopted by Riekki et al (Riekki, Salminen and Alakärppä, 2006) who describes how to access local printing and calling services. The ActivePrint project (ActivePrint 2007) also adopts preloaded clients to connect users to services which have been tagged with 2D bar codes. In this case the user takes a picture of the bar code and then she may initiate a Web download, or send an SMS or make a call or get contact data about some product. In this solution, a service middleware engine loaded in the phone activates a service stub corresponding to the tag selected. In the case of these solutions, when no i-appli or no service stub is available on the phone, no service is invoked.

The solutions highlighted above, and others that adopt pre-loaded clients to connect services to tags suffer from the fact that services can be used only when a client is actually present in the phone. To the extent that the services are well known and persistent, as in the case of accessing a bank statement or a transportation system with iMode-FeliCa, or known printing services, or SMS services with ActivePrint, the system works fine and may be extremely secure also. But in the case of advertisement posters like the one described in Figure 2 the service may be specific for the poster. In turn each poster may refer to a different service. It is unfeasible to expect that users will have all possible clients for all possible posters. Instead, we need to find a new way to provision the services associated with the posters that can flexibly connect the poster with the service without any requirement of pre-loaded clients.

A final way to connect services and tags is provided by Kerer et al. in (Kerer et al., 2007). Here RFID is used to identify objects or people that are present in a room, while Web services are used to offer services. Potentially, such a system may take advantage of the full power of SOA solutions to connect services with tags. But, despite the fact that Web services and RFID are used in the same system they perform two very different tasks and they never work together. RFID is used only for identification of people and objects in the environment. On the other side, services are invoked through the infrastructure described by the Service Oriented Architecture (SOA) (Papazoglou and Heuvel, 2007). At no time during the interaction with Web services the RFID tags are used for any other purpose than provide identification information on the objects and people in the environment. Whereas this system uses SOA, the only use of the tags is for security and permission of using services, they are not used to provision the services as would be required by an interaction with the poster in Figure 2.

In addition to the problem of the connection with services, another dimension is defined by the interaction between users and the objects. Normally, the user need to do only one action such as swiping the phone or the card in front of a tag, or take a picture of a visual marker. The action invokes the service and sends to the service the appropriate data. This user interaction paradigm is not enough to address the problem described above in relation with the poster in Figure 2, where multiple actions are required to invoke a service. To our knowledge, this problem has not been addressed in the academic literature and we are not aware of any the industrial applications that exploit this interaction mechanism. Superficially similar works have been proposed by Reilly et al in (Reilly et al., 2005) which provides a way to use complex gestures, such as “lasso select”, and by Rukzio et al. (Rukzio, Wetzstein and Schmidt, 2005) which makes a comparative study of the different types of gestures such as "pointing" to take a picture of a 2D bar code, or "touching" to read an NFC tag. Ultimately these works are orthogonal to ours since we aim at using long sequences of single gestures, while they use only one (although complex) gesture. Future work may aim at combining the different approaches.

In conclusion none of the works proposed in the literature provide a way to address the problem posed by the poster shown in Figure 2. On the service side, the proposals to connect services to tags prove to be too rigid and do not support automatic service provisioning. On the user interface side, the theories on the actions to be performed are still inadequate. The main contribution of this paper will be to overcome both limitations by providing a way to connect tags to services that support on-line service provisioning and on the user side by providing a way to support multiple actions.

THE PERCI SYSTEM

The diagram in Figure 3 provides an abstract representation of the PERCI system and its relations with the different components. As a starting point, there are three information sources: (1) a service, (2) a set of tags on the poster that refer to the service, and (3) the user who wants to take advantage of poster. To solve the interaction problem, PERCI requires a basic infrastructure which enables the interaction with these three information sources to be already present on the phone[1]. Specifically, the first component required by PERCI is the *(OWL-S) Web services stack*: a set of libraries that supports the processing of OWL-S descriptions and the automatic creation of service clients on the basis of their OWL-S specification. The second

[1] For simplicity, in this paper we use the term "phone" in a broad sense to mean the user's computational infrastructure. As discussed more in details in the implementation section, this infrastructure may include proxies running on the network. From the point of view of the user's experience there will be no difference between running code on the phone or running it on the network proxies.

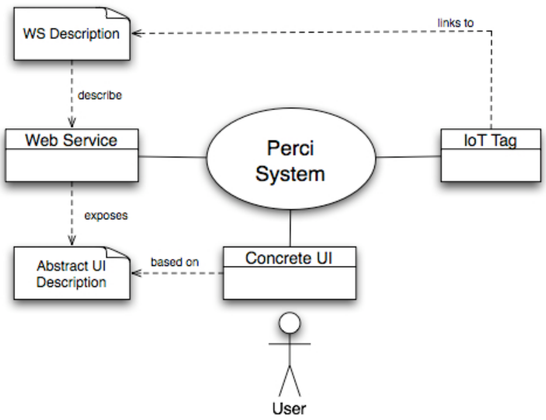

Figure 3: An abstract description of the problem

component is a *Tag Reader*, i.e. the hardware and the software that transforms the signal from the tag in digital information that the rest of the system can use. The third component is a *Rendering Engine* that transforms the abstract UI description into a concrete UI that is displayed to the user.

A crucial problem of a system like PERCI is that it mediates between the user and the service without assuming any direct relation between the reading of the tag and the invocation of the service. The problem emerges from the fact that the service follows a very well defined protocol in which the invocation of a process on the service side may require information coming from different tags; on the other hand, the user is not aware of the requirements of the service, therefore she may read the tags following a very different order. In such a case, the service and the user may loose synchronization. For example, the service may expect information in a very strict order, such as the movie theater before the movie title, whereas the user may select the movie title before selecting the theater.

To address the synchronization problem, PERCI needs to de-couple the tag reading from the service invocation to reconcile the service and the users. The resulting architecture is shown in , which highlights the main components of the PERCI system and the main data and control flow within the system. The Tag Reader is responsible for reading the tag and extracting the information. Since there are many different types of tags, such as 2D-bar codes and NFC, the Tag reader employs the PMIF framework (Rukzio, Wetzstein and Schmidt 2005) to abstract from the specific type of tag to the content of the tag which can be used within the system.

The content of the tag is then stored in the *Value Store* which is responsible to act as a buffer between the tag and the service invocation. Tags refer to the service that can process them, therefore the tag is used to instantiate the Web Service Client and the User Interface. Specifically, upon reading the tag, the

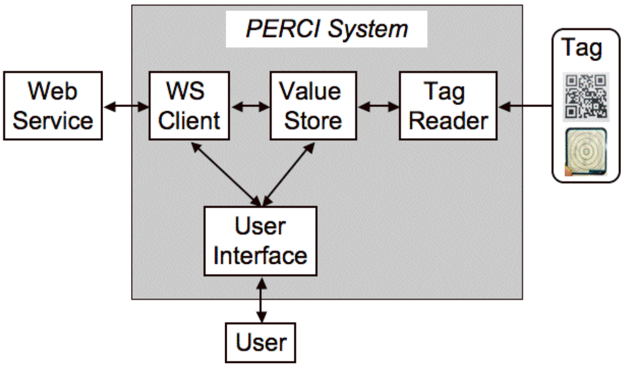

Figure 4: PERCI Infrastructure

service description is downloaded and used by the web services stack to configure on the fly the Web Service Client, and by the Rendering Engine is used to instantiate the User interface.

The centrality of the Value Store is crucial in PERCI since it guarantees the decupling between the tag reading and the service and the user interface. In broad strokes, the service client waits until all data that it expects is read by the user and put on the Value Store and only when this condition is satisfied the service is invoked. In the following three subsections we will analyze this process in greater details. Specifically we will look at the form of tags that are required, and at the interaction with the services and the user interface.

Representing the service

In order to instantiate a service client, PERCI needs to know the interaction protocol of the service, and specifically, what messages does the service does accept and in which order. In OWL-S the service interface is represented as a workflow of processes, named Process Model, in which each process is described by the input/output messages that the service exchanges with the client, and precondition/effect transformations that result from those messages. In turn the inputs and outputs are grounded into messages described using the WSDL description of the service.

In a system like PERCI, the representation of the interaction protocol is not enough, since it is crucial to interleave the interaction with the service with the interactions with the user. Users need to be sure that the phone understood their selections: that the movie that they desire is selected and that the ticket applies to the show of their choice. Furthermore, users need reassurance that the phone and the service are progressing toward a completion of the transaction. Finally, the user needs to react to failures that

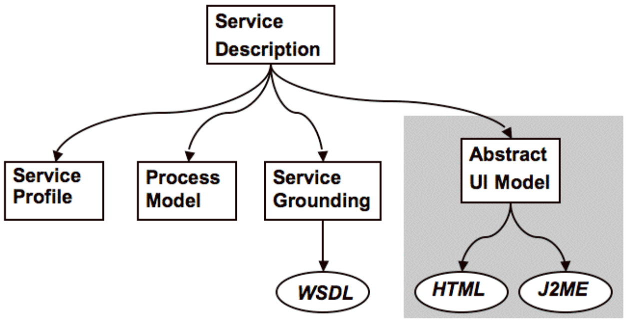

Figure 5: The Extended OWL-S Model

may occur during the interaction. For all these reasons it is essential to display a user interface that communicates to the user the status of the interaction.

The solution that we adopted to generate a user interface is to derive an abstract user interface directly from the service description (Khushraj and Lassila, 2007). Specifically, as shown in Figure 5, we extended OWL-S with a new model, which we named "Abstract UI Model", which is to be thought a parallel of the OWL-S Grounding to WSDL, but rather than mapping OWL-S processes into WSDL it maps them in a user interface. To map a process description into a user interface, the OWL-S Abstract UI Model specifies how input and output parameters map to graphic widgets, furthermore it maps the data input by the user into data that can be processed by the service. Symmetrically, it transforms output data from the service in a way that it can be exposed to the user.

One problem of associating a user interface to a mobile application like PERCI is that there is no telling on which terminal the interface will be displayed. A bottom line approach that can be adopted in this case is to assume that every mobile phone supporting data connection has a browser, so the interface is rendered in HTML. The resulting interface can be displayed on every terminal, irrespective of its capabilities; but this assumption effectively pushes the interface to the minimum lower denominator, so that it will waste the capabilities of high-end terminals or display badly on low-end terminals. The solution adopted within the PERCI system is different. Instead of committing to a technology at design and service description time, PERCI commits at execution time when the terminal to be used is known. Therefore, the graphic widgets used in the User Groundings specify how the information is presented to the user, but they do not commit to a specific technology to display the widget (Siorpaes et al., 2006). The initial set of abstract widgets used in by PERCI is shown in Figure 6. Of these widgets, *Direct Input* widget provides an arbitrary input, e.g. via a text field; *Single Select Input* provides a single value from a given choice, such as a drop down or radio button menu; *Multiple Selection Input* widget provides several values from a given choice, e.g. from a checkbox; Single Selection and Multiple Selection widgets allow a loose interpretation of the widget, meaning that a widget may be rendered as direct input or selection input; *Plain Output* parameter widget will usually be rendered as a simple textual message.

The concrete rendering is decided when the service is delivered and the terminal of the user is known. If the user terminal supports the graphic libraries of J2ME, and the terminal has sufficient resources to display a user interface, the user interface is displayed using the java graphic widgets; otherwise it is displayed using HTML.

Content of the Tags

The tags on a poster like the one shown in Figure 2 essentially provide two types of information. First, they describe the values to be assigned to the parameters. For example they say whether the movie is "Geisha" or "The Da Vinci Code". Second, they specify the function to be applied to the parameter; in other words whether the user wants to buy the ticket or whether she wants information about the movie. For this reason, PERCI distinguishes between the three types of tags.

1. *Action Tags* that are associated with processes performed by the service, such as reserving a ticket or purchasing goods. Action tags uniquely identify uniquely the process to be performed.
2. *Value Tags* that are associated with a value. For example the tags associated with the title of a movie or with the showing time. These tags contain four pieces of information: (a) the value of the data, (b) the type of the data, (c) an id which can be used to associate the data to both the information to send to the service and the information to display to the user, and (d) a URI of the service to which the data will be sent.
3. *Hybrid tags* that combine the previous two types by expressing both an action to be performed as well as a value.

The different types of tags also play a different role in the interaction with the service since action tags explicitly specify that the user wants to perform a function, such as buying tickets; while parameters tags specify the different value selections of the user. Finally, the role of hybrid tags is to provide a shortcut where the service is invoked on a given value.

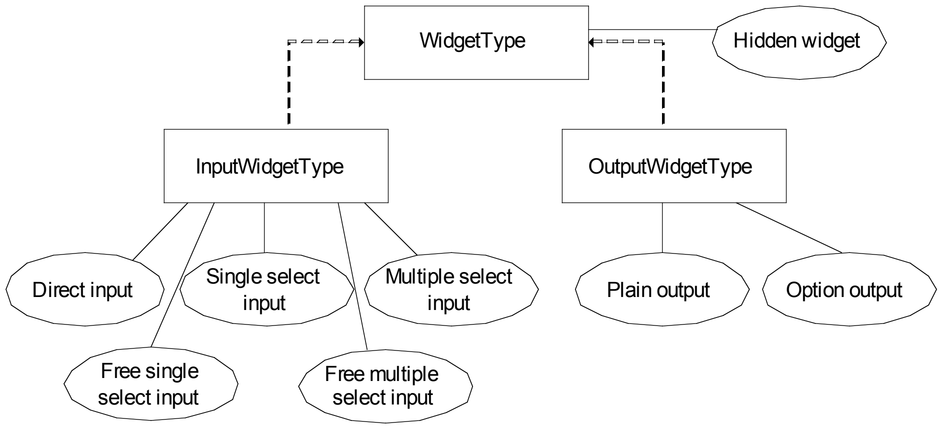

Figure 6: Types of widgets used in the abstract user interface

Managing the Interaction with the Service

Upon reading a tag, the PERCI system has to make two decisions: the first one is whether to invoke a process on the service side; the second one is whether to display a user interface.

The decision of which process to perform next is driven by the Process Model of the OWL-S service description that describes the workflow of the interaction with the service. But the decision of which process to invoke depends also on the action tags that are activated by the user and by the data that is available to send to the service. Specifically, there are three simple conditions that specify whether a process can be executed at a given time.

1. It is the next process in the order described by the protocol language;
2. The process has been selected by the user through an action tag;
3. All the input data required by the process has been provided. Such input data is provided either by the user selection of a value tag or through user input, and it should be available in the Value Store.

The third condition has an important consequence: the invocation of a process on the service is conditioned on having all data from the user. In turn, this rule prioritizes the user interface over the invocation of processes: whenever a process depends on some user data, the user interface should be displayed first. Displaying the abstract UI includes:

1. Selecting the display action to perform. This operation will result in either displaying the data, or displaying error and warning messages;

2. Selecting the data to display to the user. This selection is based on the data available in the Value Store and on the requirements of the user interface;
3. Displaying the data consistently with the Abstract User Interface specification;
4. Possibly read inputs from the user if this is required by the user interface. User inputs are then stored in the value-store.

Once all data required by the process is available in the Value Store, the Service Client is responsible for identifying which data to send to the service. Such an invocation, which is performed by the Service Client component, results in using the OWL-S Grounding to select the WSDL operation to execute and performing the data translation required to invoke the operation.

Upon completing a process invocation, the Interaction Manager waits for the results of the process. When the process completes and all the results have been received, they are added to the Value Store and possibly displayed to the user if an abstract interface is specified for that process. Finally, the Service Client moves on to the next process to execute. When the last process of the Service Process Model has been performed, the interaction with the service is completed.

Implementation

We implemented the PERCI system in a running prototype using two posters, the first one is a movie poster, the second one is a transportation poster for the Munich public transport system. In the implementation we used two types of tags. The first one based on the 2d bar codes using the Semacode toolkit (Semacode.org, 2007) running on a Nokia N70; the second one based on NFC running on a Nokia 3220 with the NFC shell. The abstraction from the tagging technology is based on PMIF (Rukzio, Wetzstein and Schmidt 2005). The Web services stack used in the implementation is based on Axis, and the Mindswap OWL-S API (Sirin, 2004) was used to execute the OWL-S description of services. We extended the OWL-S API to handle the abstract interface definition through. XML transformations were implemented on the Cocoon framework.

Given the limited memory and processing power of the NOKIA 3220, it was impossible to run the complete application on the mobile phones, therefore we had to move most of the computational load onto a proxy running on a public server. In the actual implementation the phone is responsible for the rendering of the concrete interface, while the proxy is responsible for the invocation of the services and the management of interaction proxies. The communication between the phone and the proxy was based on synchronous http calls. Since the use of a proxy in mobile applications is often considered as a "necessary evil" to overcome mobile phones limitations, three important issues need to be considered. The first one is that the synchronous http calls could be easily replaced by function calls on more powerful phones; the second issue is that many advanced service applications in the research arena require proxies, which suggests the third issue: mobile operators may devise new business models around the hosting of proxies to increase their revenue streams while supporting increasingly complex services.

DISCUSSION

Currently, the PERCI system is unique in its aim and technical nature. As discussed above, at this point, no other system features similar capabilities and functionalities in combining NFC technology with Semantic Services. In this discussion section we analyze PERCI's properties and we provide an initial evaluation of the system.

Properties of PERCI

The first property is that PERCI supports *long sequences of actions that are performed by the user*. Namely, with the different actions the user selects the movie, the theater, the time of the desired show, and the number of tickets to buy. Long sequences of actions are supported by storing the values read by the user and by passing them to the service on the bases of its needs.

The second property of PERCI is the support of *automatic invocation with arbitrary services*. This property is crucial since users of advertisement posters will not have any client specialized to interact with each poster. The process described in section 3.3 does not require any user intervention at the system level. Rather it is completely driven by the service description and by the user's reading of the tags read. Furthermore, the decoupling of tag reading and process invocation supports the separation between the user's process and the service protocol.

The third property of PERCI is the ability to *mirror the choices proposed by the poster on the mobile phone*. As a consequence the mobile phone user has the choice to either continue the interaction with the physical object, or to leave the object and continue the interaction to the screen of the phone. This feature may be very useful in case the user needs to interrupt the interaction with the object before it is concluded. Significantly, this property shows that PERCI can display the same of opening a Web page in Cooltown, ActivePrint, and 2D bar codes with iMode. The difference is that whereas in these systems, applications are designed to remove the interaction from the object pushing it to the browser. In our case, we provide two interfaces to the same object: one physical through the tags, the other virtual through the mobile's screen. The decision of which one is best is left to the user and the condition in which she is working. The user studies discussed in (Broll et al. 2006, Siorpaes et al., 2006) and summarized in Section 4.2, compare the usage of the two interaction modalities showing that by and large users prefer to interact with the object rather than shift their attention to the mobile screen.

Usability and User Experience

PERCI involves different aspects from the user interface and usability issues. To this end, we have performed two user studies (Broll et al. 2006) (Broll, 2006): the first one was a low fidelity mock-up evaluation (Nielsen, 2003) (Snyder 2003) to understand the expectations of the potential users before proceeding with the implementation. In the second study, 10 subjects were asked to use the implemented system to buy movie and transport tickets from the posters. In this study, the subjects were asked to solve the same problem under 3 different conditions. In the first condition the subjects interacted with the posters using only NFC tags; in the second condition the subjects were required to use only direct input on the phone through menus and by typing implementing the interaction mechanism employed by systems such as iMode, Active Print and Cooltown described above; in the third condition, the subjects were required to use 2D bar codes instead of NFC tags.

In summary, the results of both studies have been positive and promising especially given that most European mobile users are not yet used to NFC and tagging technology in general. The detailed results of the second experiment are shown in Table 1**Error! Reference source not found.**. The table is organized in three rows describing the three conditions (namely: the use of NFC tags, the use of Direct Input to simulate existing systems, and the use of 2D bar codes). The users were asked to evaluate whether they found the system *easy* to use, *fun* and appealing, and whether they had the feeling that it was *reliable* and they would trust such a system. As **Error! Reference source not found.** shows the subjects found the NFC and Direct Input versions of the system easy and reliable. This result is quite interesting since the subjects had great familiarity with the use of menus and forms on the Web and other applications therefore they were very familiar of the "direct input" condition, on the opposite of the NFC condition since they used NFC for the first time. This suggests that NFC can be easy to use in practice. On the other side, the use of NFC was found more fun and appealing of the use of "direct input". 2D bar codes scored quite badly in the experiment because their use is more cumbersome of NFC tags and resulted in a negative

		Applies	Not Applies	Unsure
NFC	Easy	10	0	0
	Fun	8	1	1
	Reliable	9	1	0
Direct Input	Easy	8	0	2
	Fun	0	3	7
	Reliable	10	0	0
2D bar codes	Easy	0	9	1
	Fun	3	7	0
	Reliable	4	5	1

Table 1: Results of the Experiment

impression on the users. In conclusion, the experiment revealed that systems based on the multiple touch ideas presented in this paper may be very appealing to mobile users, and that the PERCI system may result in very usable applications in the real World.

While the PERCI system provides an initial solution to the problem of interaction with advertisement posters, and potentially with arbitrary objects, it also raises a number of problems that were unseen before. On the user side, the user interface is split between the poster and the mobile phone, which inevitably means that the user needs to shift attention between the two interfaces. The reduction and possibly the management of such attention shifts is crucial for the success of this technology since it entails a cognitive overload that may vanish the advantages of the technology. Furthermore the design of the poster and of the object in general becomes crucial since the object needs to achieve different goals in the same time and satisfy very different constraints. For example, an advertisement poster should be appealing to entice users to buy from it, but also easy to use with the tags placed in a convenient and intuitive way, and it should satisfy geometric constraints on the size of the icons to be presented: typically if the icons become too small and too close two or more icons may be served by the same tag making them indistinguishable. An additional problem is the management of wrong or inconsistent user actions, such as double selections of the same tag, or missed selections of tags, such as missed selection of the movie theater, or showing time

Figure 7: A PERCI Poster and an Equivalent Ticketing Machine

CONCLUSIONS

Tagging technology is bound to become a common part of our life. In a short while many usual actions, from opening a door, to paying for a bus ride will be performed using tags and possibly our mobile phone. Virtually every object then will become a source of electronic information. Although, this vision may sound very futuristic, the transformation is already happening as shown by the poster in Figure 1 and by the growing applications of NFC and 2D bar codes that are emerging on the market. Somewhat surprisingly, within the Semantic Web there has been no appreciation of these events and virtually no work toward the semantic annotation of tags.

To our knowledge, PERCI is the first attempt to bring semantics to tags and to use them to invoke web services. As a result PERCI opens new opportunities that go beyond what has been deployed as existing services such as the linking of 2D bar codes and basic Internet-based i-mode services in Japan and existing ticketing solutions. Specifically, PERCI shows how exploiting semantic web service technology it is possible to explore new types of relations between objects and services on one side and more complex interactions between users and services.

The impact of PERCI-like systems is potentially huge. Consider the poster displayed on the left side of Figure 7; it has the same functionalities of the ticketing machine displayed on the right side of the same figure, but with two striking differences. First, the cost of the poster is much lower (approximately 50 €) with virtually no maintenance costs; second, additional information can be loaded on the phone that cannot be loaded on a ticket, as for example the way to go to a given station.

From the SOA point of view, PERCI opens a new way to provision services to users, and therefore also a new way to think about service composition. The traditional SOA-based composition assumes universal availability of services through the network, but PERCI shows that there are other ways to provision services that break that assumption. Indeed services are not universally available, but only available in the short range of the tags that are reachable by the user. Semantics is going to play a pivotal role in the provisioning of this type of pervasive services

The fundamental question raised by the paper is how to combine services offered to the user, keeping into account that pervasive services are deployed in different places with different modalities and their provisioning to the user is necessarily spread over a long time. In such a case, the tag read by a poster may be used in new and unexpected ways to work with a service for which they were not designed. The solution of this problem requires semantic annotation both in the abstraction of the content of the tags to find the most appropriate service to process them, as well as in the invocation of those services. PERCI is a first step toward this new model of opportunistic service composition in which services may be combined on the fly depending on the users' needs, and her context.

REFERENCES

ActivePrint (2007). *Active Print Web Site*. Retrieved Sept 13, 2007, from http://www.activeprint.org/

Adams, R. (2007, April 04) *2-D Bar Code Page*. Retrieved September 11, 2007, from http://www.adams1.com/pub/russadam/stack.html

Deutsche Bahn AG (2007). *Handy-Ticket*. Retrieved Sept 11, 2007; www.bahn.de/p/view/planen/reiseplanung/mobileservices/handy_ticket.shtml

G. Broll, Siorpaes, S. Rukzio, E. Paolucci, M. Hamard, J. Wagner, M. Schmidt., A. (2006) *Supporting Service Interaction in the Real World*; In Permid Workshop in conjunction with Pervasive 2006, Dublin, Ireland, May 7 2006.

Broll, G. (2006); *Mobile Interaction with Real World Services*; Diploma Thesis, Institut für Informatik, Ludwig-Maximilans-Universität München; July 2006.

Chinnici R., Moreau J., Ryman A. *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. W3C Candidate Recommendation 27 March 2006; http://www.w3.org/TR/2006/CR-wsdl20-20060327

ECMA International (Sept. 2006). *Near Field Communication White paper*; Ecma/TC32-TG19/2004/1

EPCGlobal *RFID Implementation Cookbook, 2nd Release*. Retrieved September 11, 2007, from http://www.epcglobalinc.org/what/cookbook/

Hazas, M. Scott, J. Krumm, J. (2004) *Location-aware computing comes of age*. Computer; Volume: 37, Issue: 2; pages 95- 97; 2004

IDTechEx (2007); *RFID Logistics Case Studies: Thirty detailed RFID logistics case studies*. From http://www.idtechex.com/products/en/view.asp?productcategoryid=49. Retrieved September 22, 2007,

Kerer, C., Dustdar, S., Jazayeri, M, Gomes, D., Szego, A., Burgos Caja, J.A. (2004). *Presence-Aware Infrastructure using Web services and RFID technologies*. 2nd European Workshop on Object Orientation and Web Services, ECOOP Workshop, 14 June 2004, Oslo, Norway, Springer LNCS

Khushraj, D., Lassila, O. (2005) *Ontological Approach to Generating Personalized User Interfaces for Web Services*. 4th International Semantic Web Conference (ISWC 2005)

Kindberg T. and Barton, J. (2001); *A Web-based nomadic computing system*; Computer Networks; Volume 35, Issue 4, March 2001, Pages 443-456

Kindberg, T. et al; People, Places, Things: Web Presence for the Real World; Journal Mobile Networks and Applications; Volume 7, Number 5 / October, 2002, Pages 365-3761

Martin, D. Burstein, M. McDermott, D. McIlraith, S. Paolucci, M. Sycara, K. McGuinness, D. L. Sirin E. and Srinivasan, N. (2007) *Bringing Semantics to Web Services with OWL-S*. World Wide Web. Volume 10, Number 3 / September, 2007

Meloan, S. (2003) *Toward a Global "Internet of Things"*. Sun Developers Network. November 11, 2003. Last Reviewed on September 22, 2007 from http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/

Nielsen, J. (2003) *Paper Prototyping: Getting User Data Before You Code*. Last Reviewed on September 22, 2007 from http://www.useit.com/alertbox/20030414.html

Papazoglou, M. Heuvel, W. (2007) Service oriented architectures: approaches, technologies and research issues. The VLDB Journal. Volume 16, Issue 3 (July 2007)

Riekki, J. Salminen, T. Alakärppä, I. (2006) *Requesting Pervasive Services by Touching RFID Tags*. IEEE Pervasive computing, 2006

Reilly, D., Dearman, D., Welsman-Dinelle, M. and Inkpen, K. M. (2005). Evaluating Early Prototypes in Context: Trade-offs, Challenges, and Successes. IEEE Pervasive Computing (October-December 2005), 4(4):10-18.

Rukzio, E. Wetzstein, S. Schmidt A. (2005) *A Framework for Mobile Interactions with the Physical World*. Wireless Personal Multimedia Communication (WPMC'05). Aalborg, Denmark, 2005.

Semacode.org (2007). *Semacode SDK technical paper*; Retrieved September 21, 2007, from http://semacode.org/about/technical/

Siorpaes, S. Broll, G. Rukzio, E. Paolucci, M. Hamard, J. Wagner, M. Schmidt, A. *Supporting Service Interaction in the Real World*. Semantic Desktop Workshop 2006

Siorpaes, S. (2006) *A Physical Mobile Interactions Framework based on Semantic Descriptions*. Diploma Thesis, Institut für Informatik, Ludwig-Maximilans-Universität München; July 2006.

Sirin E. (2004) OWL-S API. Last Reviewed on Sept 22, 2007, From http://www.mindswap.org/2004/owl-s/api/

Snyder, C. (2003) *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. The Morgan Kaufmann Series in Interactive Technologies; 2003

Tohamy, N. (2005) *The Present and Future of RFID in Logistics*. Forrester Research. From http://www.rfidupdate.com/articles/?id=998; on Wednesday November 16th, 2005

Toye, E. et al (January 2007). Interacting with mobile services: an evaluation of camera-phones and visual tags; in Personal and Ubiquitous Computing; Volume 11; Issue 2; Pages: 97 - 106

UsingRFID.com (2007); *UK-based NFC smart poster marketing system launched*; http://www.usingrfid.com/news/read.asp?lc=h33830ax1078zd; Monday May 14, 2007

Want, R. et al (1999) Bridging physical and virtual worlds with electronic tags; In Proceedings of the Conference on Human Factors in Computing Systems (CHI99); Pages: 370 - 377

Yoshinaga H., Hattari Y., Sato T., Yoschida M. and Washio S. (2003) *i-Mode FeliCa*. NTT DoCoMo Technical Journal, Vol 6, No 3, Dec 200