Semantic Web Technologies for Context-Aware Museum Tour Guide Applications

Shih-Chun Chou[1], Wen-Tai Hsieh[1], Fabien L. Gandon[2] and Norman M. Sadeh[2*]

[1]Advanced e-Commerce Institute
Institute for Information Industry
Taipei, Taiwan, ROC

[2]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891 - USA

Abstract

Traditionally, visitors to museums have been left having to choose between finding their way around exhibits on their own or taking a standardized group tour with a guide. In this paper, we describe a context-aware museum tour guide that adjusts its recommendations to the interests and contexts of individual visitors and enables them to selectively share their experience with others. The tour guide is built around an innovative Semantic Web framework that minimizes the development and maintenance costs associated with the introduction of new exhibits, new visitor-oriented services and new sources of contextual information. In particular, it features a semantic web rule reasoning engine that enables visitor-oriented services to identify relevant sources of contextual information and to enforce user-specified privacy preferences about what information they are willing to share with others (e.g. "Only members of my group can see my current location", or "Only my friends can see how I rate exhibits"). While still in prototype stage, the tour guide's target environment is the National Museum of Natural Science, one of Taiwan's largest museums with over 3 million visitors per year..

1. Introduction

The traditional museum visitor experience has been characterized by having to choose between a limited number of predefined guided tours and the challenge of visiting on one's own. In this paper, we detail work on a knowledge-rich, context-aware museum tour environment aimed at ***providing visitors with a more engaging and more personalized experience*** [1]. The environment revolves around a Semantic Web framework built to capture and exploit knowledge about the museum's exhibits, its layout as well as contextual information about visitors (e.g. their interests, their locations, how much time they have available, whether they are visiting as part of a group, etc.) and relevant privacy preferences. We discuss key features of our architecture, focusing on how they improve over prior efforts to develop context-aware museum tour guides and how they also extend earlier Semantic Web research. In particular, our use of Semantic Web technologies facilitates the development and refinement of new domain ontologies, whether to describe new exhibits, visitor interests or new sources of contextual information. Our reasoning technology allows for the definition of rules that refer to these ontologies, whether to specify ***application-specific preferences*** (e.g. "If I have some extra-time, please tell me about places where I can eat at the museum") or ***privacy preferences*** (e.g. "Only show my location to my teacher").

As they enter the museum, visitors are given a PDA that they carry with them for the duration of their tour. As a first step, the PDA asks them a few simple questions to understand their interests and relevant preferences, including some privacy preferences. This process has been kept as light as possible – to minimize burden on the user. It only takes a couple of minutes. Based on this information, which includes information about the user's interests, age, and how much time he or she has available, the guide builds an initial tour. This tour is used to give directions to the visitor and is adjusted, as needed, as the tour progresses. For instance, if the visitor falls behind, the tour guide will tell him so and may recommend dropping one or more exhibits from the current tour. On the other hand, if the visitor has extra time, the guide may suggest additional exhibits or possibly offer the user to take a break for a snack at a nearby restaurant.

Our museum tour guide is not built as a single monolithic application. Instead, it is an environment aimed to support a growing collection of applications that share access to information

[*] Corresponding author: Norman Sadeh (sadeh@cs.cmu.edu)

about the visitor's interests and context subject to individual privacy preferences. This includes both pull and push applications. An example of a push application can come in the form of suggestions sent to a visitor recommending that he or she takes a look at a nearby exhibit. An example of an equivalent pull application is one where the visitor can ask for a similar recommendation. The environment also supports peer-to-peer interactions between users, enabling them to find one another and share comments and ratings about different exhibits.

Within our architecture, each user has an eWallet that serves as a clearinghouse and gatekeeper to resources available for that user. This includes both information resources about his interests, contextual attributes and exhibition ratings/comments as well as communication resources such as a display for push messages. Applications that seek to access any of this information submit queries to the visitor's eWallet. The eWallet checks the user's privacy preferences to determine whether the query is compatible with the user's preferences and, if it is, returns an answer or activates a service (e.g. displaying a push message). Privacy preferences currently supported by the eWallet include ***access control preferences*** (e.g. "Only members of my class can see my location"), ***obfuscation preferences*** (e.g. "Members of my group can see the exhibit I am at but other visitors can only see the building I am in"). They can also include ***disruption preferences*** (e.g. "Don't send me promotional messages about items for sale at the museum store").

Museum tour guide applications as well as visitor eWallets are implemented on top of a ***Semantic Web Rule Engine (SWRE)*** that generalizes some of our earlier work in MyCampus, a semantic web environment aimed at enhancing everyday campus life with context-aware applications [5]. This architecture would make it possible to eventually extend the current museum tour guide environment to other tourism, education or sightseeing applications, enabling these different applications to leverage a common set of contextual resources and user preferences.

The remainder of this article is organized as follows. Section 2 provides a brief literature review. Section 3 outlines the architecture of our Semantic Web environment for context-aware museum tour guide applications. Section 4 briefly outlines relevant ontologies and rules and describes our Semantic Web Rule Engine. Section 5 outlines scenarios aimed at illustrating the operation of our environment. Section 6 provides a summary along with some concluding remarks.

2. Brief Literature Review

Prior efforts to develop context aware applications are many. Early work in context awareness includes the Active Badge System developed at Olivetti Research Lab to redirect phone calls based on people's locations [15]. The ParcTab system developed at the Xerox Palo Alto Research Center in the early nineties relied on PDAs to support a variety of context-aware office applications (e.g. locating nearby resources such as printers, posting electronic notes in a room, *etc.*) [Schi95, Schi94]. Other relevant applications that have emerged over the years range from location-aware tour guides (e.g. HIPS [11]) to context-aware memory aids. More recent research efforts in context awareness include MIT's Oxygen [3], CMU's Aura [7] and several projects at Berkeley's GUIR (e.g. [8]) to name just a few.

While early context-aware applications relied on *ad hoc* architectures and representations, it was quickly recognized that separating the process of acquiring contextual information from actual context-aware applications was key to facilitating application development and maintenance. Georgia Tech's Context Toolkit represents the most significant effort in this direction [DA99, DSFA00]. In the Context Toolkit, widgets act as wrappers that provide access to different sets of contextual information (e.g. user location, identity, time and activity), while insulating applications from context acquisition concerns. Each user (as well as other relevant entities such as physical objects or locations) has a context server that contains all the widgets relevant to it. This is similar to our notion of e-Wallet, which serves as a directory of all personal resources relevant to a given user (e.g. relevant location tracking functionality, relevant collections of preferences, access to one or more calendar systems, *etc.*). Our Semantic e-Wallet however goes one step beyond Dey's Context Toolkit. It makes it possible to leverage much richer models of personal resources - what personal information they give access to, when to access one rather than the other, how to go about accessing these resources. In addition, it includes access control and obfuscation functionality to enforce user privacy preferences.

This richer model was first demonstrated in MyCampus, a Semantic Web environment for enhancing everyday campus life through an open collection of context-aware applications [SCV+03, GS04]. Our notion of Semantic eWallet also extends current eWallet technologies by providing a unified privacy front-end to all the resources available for a given user. It also extends work on languages such as P3P/APPEL [26], EPAL [22], SAML) [16], or XACML [17] by allowing for the definition of policies that refer to an open collection of ontologies and by allowing for context-sensitive policies, including obfuscation policies.

3. System Architecture

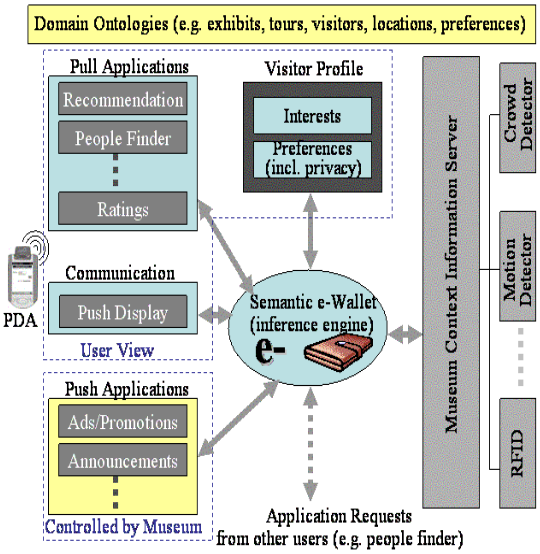
Domain Ontologies (e.g. exhibits, tours, visitors, locations, preferences)
Pull Applications
Recommendation
People Finder
Ratings
Visitor Profile
Interests
Preferences (incl. privacy)
Communication
Push Display
PDA
User View
Semantic e-Wallet (inference engine)
e-
Push Applications
Ads/Promotions
Announcements
Controlled by Museum
Application Requests from other users (e.g. people finder)
Museum Context Information Server
Crowd Detector
Motion Detector
RFID

Figure 1 Museum Guide Architecture

Each user interacts with an open collection of applications via a PDA he receives upon entering the museum. As a first step, the user is asked a few questions aimed at identifying his interests as well as relevant preferences, including privacy preferences. This information is stored in a visitor profile (Figure 1). Applications available to the user include both pull and push applications. An example of a pull application is a recommendation service to suggest additional exhibits to the user, based on his interests, exhibits he has already seen, his current location, and how much time he still has available. Other examples include a service to provide users with directions on how to go to a particular exhibit, a people finder to locate other visitors, a rating application that enables visitors to share ratings and comments about exhibits. Examples of push applications include public announcement messages (e.g. shop closing times, movie start times, museum closing times, etc.) and possibly ads/promotions (e.g. telling a visitor that a book about an exhibit he is visiting is available at the museum store). Requests to send push messages to the user or requests to access information about the user (e.g. his preferences or one or more of his contextual attributes such as his location) are directed to a Semantic eWallet. The Semantic eWallet, which is described in more detail in Section 4 and 5, is a ***reasoning engine*** that enforces the user's privacy preferences. This includes enforcing preferences about who/what application has the right to access what information about the user (and at what level of granularity) and what types of push messages the visitor is willing to accept. The visitor's interests and his preferences, including privacy preferences, are expressed in relation to concepts defined in OWL ***domain ontologies*** [16]. The visitor's profile, his push display application as well as the infrastructure used to monitor the user's context are represented as web services that can be invoked by the Semantic eWallet. As such the Semantic eWallet serves as a ***unified front-end to resources available for the visito*r**, acting both as a ***clearinghouse and a gatekeeper to these resources***.

Details about the infrastructure used to monitor and access contextual information are provided in [10]]. It revolves around an SQL query interface to an open collection of sensor services. Current sensor services include RFID tags and readers used for people location tracking. Motion detectors are used to monitor visitor streams. The infrastructure also supports higher level services such as services to compute the distance from a visitor's current location to a particular exhibit.

4. Semantic Web Technologies for Privacy and Service Invocation

Applications, user profiles and contextual information in our museum tour guide environment refer to an open collection of ontologies expressed in OWL [16]. These ontologies define relevant domain concepts such as exhibits, tours, exhibit topics, contextual attributes, etc. For instance, Figure 2 provides a simple definition of the concept of museum exhibit in OWL and Figure 3 shows an instance of this concept (in RDF).

We have also developed ROWL, an extension of OWL, to specify rules that relate to concepts defined in our ontologies [6]. In ROWL, a rule is expressed as consisting of a *body* (i.e. a set of conditions) and a *head* (i.e. an action to be taken if the conditions are satisfied). Rules in ROWL can include variables that refer to instances of concepts defined in ontologies. A simple example of a rule is illustrated in Figure 4, indicating the preference of a particular visitor, Ben, to have an extra stop inserted in his tour, if his "early_tardy_status" is "early", namely if he has some extra time. A variation of this rule for another visitor could indicate that he would rather stop at a restaurant and eat, if he has some time left.

```
<owl:Class rdf:ID="Exhibit"/>
<owl:DatatypeProperty rdf:ID="name">
      <rdfs:domain rdf:resource="#Exhibit"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="topic">
      <rdfs:domain rdf:resource="#Exhibit"/>
      <rdfs:range rdf:resource="http://iii.org.tw/ontology/Topic#Topic"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="recommended_duration_time">
      <rdfs:domain rdf:resource="#Exhibit"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="CrowdSize">
      <rdfs:domain rdf:resource="#Exhibit"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="TargetAgeGroup">
      <rdfs:domain rdf:resource="#Exhibit"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="interactivityLevel">
      <rdfs:domain rdf:resource="#Exhibit"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</owl:DatatypeProperty>
```

Figure 2. Simplified concept of museum exhibit.

```
<exhibit:Exhibit rdf:ID="exhibit_Ofd10231">
      <exhibit:name>Our Galaxy</exhibit:name>
      <exhibit:topic rdf:resource="&topicins;#astronomy 124"/>
      <exhibit:recommended_duration_time>10</exhibit:recommended_duration_time>
      <exhibit: CrowdSize >nil</exhibit: CrowdSize>
      <exhibit:TargetAgeGroup>elementary school</exhibit:TargetAgeGroup>
      <exhibit: interactivityLevel >for watch</exhibit: interactivityLevel >
</exhibit:Exhibit>
```

Figure 3. An instance of a museum exhibit.

Our rule ontology includes more specialized rules such as privacy rules and service invocation rules. An example of a privacy rule is illustrated in Figure 5. Privacy rules are defined using three tags [5]: a *content* tag describes the piece of knowledge to which the privacy rule applies (e.g. a visitor's location); a *check* tag specifies conditions that need to be met to authorize read access (e.g. to allow someone to see the visitor's location), and an optional *revision* tag is used to specify possible obfuscation rules, e.g. to specify that location information can only be disclosed at the level of the building the user is in but not at the level of the particular exhibit he is at. Figure 5 illustrates on such rule where someone only allows his teammates to see his location at the level of the exhibit ("stop") he is at.

```
<rowl:Rule rdf:ID="rule1">
      <rdfs:label>Insert a new exhibit</rdfs:label>
      <rowl:body rdf:parseType="Collection">
         <visitor:Visitor rdf:about="#VisInst">
            <visitor:name>Ben</visitor:name>
            <visitor:tour rdf:resource="#tour"/>
            <visitor:interest rdf:resource="#pref"/>
            <visitor:extra_time_preference>visit</visitor:extra_time_preference>
            <visitor:early_tardy_status>early</visitor:early_tardy_status>
         </visitor:Visitor>
         <exhibit:Exhibit rdf:about="#e">
            <exhibit:topic rdf:resource="#pref"/>
         </exhibit:Exhibit>
         <tour:Tour rdf:about="#tour">
            <tour:stop>
                  <stop:Stop rdf:about="#stop">
                        <stop:nextstop>nil</stop:nextstop>
                  </stop:Stop>
            </tour:stop>
         </tour:Tour>
         <stop:Stop rdf:about="#stop1">
            <stop:stop_at rdf:resource="#e"/>
         </stop:Stop>
      </rowl:body>
      <rowl:head rdf:parseType="Collection">
         <tour:Tour rdf:about="#tour">
            <tour:stop>
                  <stop:Stop rdf:about="#stop">
                        <stop:nextstop rdf:resource="#stop1"/>
                  </stop:Stop>
            </tour:stop>
            <tour:stop>
                  <stop:Stop rdf:about="#stop1">
                        <stop:stop_at rdf:resource="#e"/>
                        <stop:previousstop rdf:resource="#stop"/>
                  </stop:Stop>
            </tour:stop>
         </tour:Tour>
      </rowl:head>
</rowl:Rule>
```

Figure 4 – Expressing visitor preferences as ROWL rules.

```
<sowl:PrivacyRule>
   <rdfs:label>people can only know the Location of teammate</rdfs:label>
   <sowl:target>
      <visitor:Visitor rdf:about="&variable;#VisInst">
         <stop:Stop_at rdf:resource="&variable;#stop_at"/>
      </visitor:Visitor>
   </sowl:target>

   <sowl:check>
      <rowl:And>
         <rowl:condition>
              <qowl:Query rdf:about="&variable;#query">
                 <qowl:sender rdf:resource="&variable;#teammate" />
              </qowl:Query>
         </rowl:condition>
         <rowl:condition>
              <visitor:Visitor rdf:about="&variable;#VisInst">
                   <visitor:Teammate>
                        <visitor:visitor rdf:resource="&variable;#teammate"/>
                   </visitor:Teammate>
              </visitor:Visitor>
         </rowl:condition>
      </rowl:And>
   </sowl:check>

   <sowl:revision>
      <visitor:Visitor rdf:about="&variable;#VisInst">
            <stop:Stop_at rdf:resource="&variable;#stop_at"/>
      </visitor:Visitor>
   </sowl:revision>
</sowl:PrivacyRule>
```

Figure 5. Example of a Privacy Rule.

Within our environment, queries supported by the Museum Context Information Server as well as resources such as displaying push messages on the visitor's PDA are modeled as web services that can dynamically be invoked by rules. For instance, a request to display a promotion message on the visitor's PDA is directed to that visitor's eWallet. Upon receiving such a request, the eWallet checks the request against the visitor's privacy preferences (in this case disruption preferences). If the request is compatible with the visitor's preferences, it gets displayed on his PDA by invoking the push display application as a service. Similarly, rules that require information about the user's context such as his location result in the invocation of services supported by the Museum Context Information Server.

Technology to reason about facts and rules has been implemented in two different forms:

- A generic ***Semantic Web Rule Engine (SWRE)*** that can be used as a basis for developing tour guide applications (e.g. an application to recommend additional stops to the visitor using rules such as the one illustrated in Figure 4)
- A ***Semantic eWallet***, which provides a unified front-end to resources available for a given visitor subject to his privacy preferences. This includes access to the visitor's contextual information and preferences as well as access to his PDA to display push messages. The Semantic eWallet can be viewed as a specialized version of the SWRE that supports both privacy and service invocation rules.

Both reasoning engines have been implemented in JESS [9], using XSLT transformations to translate RDF/RDF-S, OWL and ROWL annotations into CLIPS [5]. The engines also support rollback points that make it possible to retract and modify facts (e.g. removing a stop from a visitor's tour).

By relying on an open collection of ontologies, by modeling preferences as rules expressed in terms of these ontologies and by modeling contextual resources and applications as web services, our museum tour guide environment makes it easy to introduce new exhibits, new preferences, new sources of contextual information and new applications.

5. Illustrative Scenarios

Visitors are insulated from the complexity of the underlying Semantic Web infrastructure by simple menu-based interfaces. Upon entering the museum, they are prompted to specify their interests by selecting from a list of topics. They are also asked to indicate how much time they have available and to specify a few simple preferences, such as whether they plan to eat at the museum. Privacy preferences are currently specified using three slider bars: one for museum-operated push services, one for visitors that are part of the same group, and one for other visitors. Each slider bar requires visitors to select from three predefined privacy profiles: low, medium or high. Once selected, these options are translated into rules and facts that refer to the museum's ontologies and are stored in the visitor's profile (Figure 1). Based on this profile, an initial tour is developed for the visitor. The tour is used to provide directions to the visitor and is updated during the course of his visit based on contextual information such as whether the user has fallen behind or based on crowds at different exhibits. The following illustrates two scenarios supported in the current prototype implementation.

5.1 People Finder Application

Consider two visitors, Ben and Ted, who are part of a same group. They have different interests. Accordingly, their individual museum tour guides have developed different tours for them. Ted has specified that he is willing to let members of his group see his location. This preference has been translated into a rule similar to the one shown in Figure 5 and is stored in his individual profile. As lunch time is drawing near, Ben decides to look up his friend, Ted, using the People Finder application available as a button on the screen of his PDA. Ben's People Finder application sends a request to Ted's e-Wallet, which in turn checks it against Ted's privacy preferences. The eWallet determines that Ben is a member of Ted's group and that therefore he is allowed to see his location. The eWallet proceeds and invokes the Museum Context Information Server, requesting Ted's location. Upon receiving the location back, the eWallet forwards it to Ben's PDA. Ben asks his PDA to give him directions to Ted's current location so that the two of them can go and have lunch together.

5.2 Dynamic Recommendation Application

Ted and Ben end up having a rather quick lunch together. As he resumes his tour, Ted is told that he has some extra time and is asked whether he would like a recommendation for another exhibit he might want to look at. Ted accepts, which triggers his recommendation application to look for a nearby exhibit that matches his interests and does not take longer than the extra time he has available. This recommendation involves accessing a special service supported by the Museum Context Information Server to compute the time it takes to walk between different places.

6. Summary and Concluding Remarks

In this paper, we introduced a context-aware museum tour guide that adjusts its recommendations to the interests and contexts of individual visitors and enables them to selectively share their experience with others. The tour guide is built around an innovative Semantic Web framework that minimizes the development and maintenance costs associated with the introduction of new exhibits, new visitor-oriented services, new sources of contextual information and new preferences. In particular, it features a semantic web rule reasoning engine that enables visitor-oriented services to invoke relevant sources of contextual information and to enforce user-specified privacy preferences (e.g. "Only members of my group can see my current location", or "Only my friends can see how I rate exhibits" or "I don't care for promotional messages"). While still in prototype stage, the tour guide's target museum is the National Museum of Natural Science in Taiwan, a museum that welcomes over 3 million visitors per year.

7. Acknowledgements

This research was in part supported by the Institute for Information Industry and the Ministry of Economic Affairs, Department of Industrial Technology. Part of this work has also been conducted under the DAML research initiative and has been sponsored by the Air Force Research Laboratory under contract F30602-02-2-0035. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

8. References

[1] Cooper, Jonathan. "Engaging the [Museum] Visitor: Relevance, Participation & Motivation in Hypermedia Design". International Conference on Hypermedia and Interactivity in Museums (ICHIM), Cambridge, England, 1993. http://www.artgallery.nsw.gov.au/staff/jcooper/museum_education/engaging

[2] Dey, A.K., Abowd, G.D.: Toward a Better Understanding of Context and Context-Awareness, GVU Technical Report GIT-GVU-99-22. College of Computing, Georgia Institute of Technology, (1999)

[3] Dertouzos, M.: The Future of Computing, Scientific American, August (1999)

[4] . Dey, A., Salber, D., Futakawa, M., Abowd, G.: An Architecture to Support Context Aware Computing, Technical Report GIT-GVU-99-23. College Computing, Georgia Institute of Technology, Nov. (2000)

[5] Gandon, F. and Sadeh, N., "Semantic Web Technologies to Reconcile Privacy and Context Awareness", Web Semantics Journal. Vol. 1, No. 3, 2004. Also available as School of Computer Science Technical Report CMU-ISRI-03-107.

[6] F. Gandon, and N. Sadeh. ROWL: Rule language in OWL and translation engine for JESS. Mobile Commerce Laboratory, Carnegie Mellon University,2004. http://mycampus.sadehlab.cs.cmu.edu/public_pages/ROWL/ROWL.html

[7]Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste P.: Project Aura: Towards Distraction-Free Pervasive Computing, IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments, Vol. 1, Number 2, April-June, 2002.

[8] Hong, J., Llanday, J.: A Context/Communication Information Agent, in Personal and Ubiquitous Computing Special Issue Situated Interaction and Context-Aware Computing, Vol. 5(1) (2001) 78-81

[9] Ernest Friedman-Hill, "JESS IN ACTION, Rule-Based Systems in Java," Manning Publications Co. 1999.

[10] N. Miller, G. Judd, U. Hengartner, F. Gandon, P. Steenkiste, I. H. Meng, M. W. Feng, and Norman Sadeh. "Context-aware computing using a shared contextual information service". "Hot Spots", Pervasive 2004, April 2004, Vienna, *Advances in Pervasive Computing*, Austrian Computer Society (OCG), ISBN 3-85403-176-9,

[11] Opermann, R., and Specht, M. (2000). A Context-sensitive Nomadic Information System as an Exhibition Guide. Proc. of 2nd Intl. Symposium on Handheld and Ubiquitous Computing, Bristol.

[12] Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. Proc. of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, Santa Cruz, CA, (1994) 85-90

[13] Schilit, W.: A System Architecture for Context-Aware Mobile Computing, Ph.D. Thesis, Columbia University, 1995.

[14]Sadeh, N., Chan, T-C, Van, L., Kwon O., and Takizawa, K. "Creating an Open Agent Environment for Context-Aware m-Commerce", in "Agentcities: Challenges in Open Agent Environments", Ed. by Burg, Dale, Finin, Nakashima, Pagham, Sierra and Willmott, LNAI, Springer Verlag, pp. 152-158, 2003.

[15] Want, R., Hopper, A., Falcao, V., Gibbons, J.: The Active Badge Location System, ACM Transactions on Information Systems 10(1) (1992) 91-102.

[16] OWL Web Ontology Language Reference, Working Draft 31 March 2003, http://www.w3.org/TR/owl-ref/