A Unified Architecture for Supporting Direct Tag-Based and Indirect Network-Based Resource Discovery

Simone Mora[1] and Babak A. Farshchian[2]

[1] Norwegian University of Science and Technology
[2] SINTEF ICT
Trondheim, Norway
Simone.Mora@idi.ntnu.no,
Babak.Farshchian@sintef.no

Abstract. Discovering and integrating ambient computational resources is a central topic in AmI. There are two major existing approaches: indirect network-based resource selection and direct tag-based resource identification. We motivate the need to integrate the two approaches through a scenario. We then present an architecture for a pluggable discovery system called UbiDisco. We demonstrate how UbiDisco implements a seamless integration of the two approaches at user interaction level through a framework for implementing discovery actions.

1 Introduction

AmI environments often contain embedded resources that can be used to enhance the experience of the user, or to improve the performance of the underlying system in one way or another. In order to allow dynamic and unobtrusive access to resources in the environment, applications and services need to reconfigure continuously based on the current context of the users as the users move around. As the number of network-connected resources grows it becomes quite time-consuming for users and system administrators to manage meaningful connections among resources. The trade-off between the effort to find and connect to the right resources in one hand, and the perceived value gained from these resources on the other hand, is a central research question for us.

Automation of resource associations in an AmI system requires a first essential step, i.e. discovery and integration of the resource into the AmI system. Dynamic resource discovery systems have therefore shown to be a major cornerstone of all AmI and ubicomp systems [3]. Conventional resource discovery architectures have been used to connect computers. They are therefore implemented with network topology in mind. AmI systems require a higher degree of user-centeredness. It is the intention of the user that is in focus when discovering new resources, and not the network topology. A recent line of research has therefore focused on supporting direct user-initiated integration of resources into AmI systems using embedded tags [4][5].

B. de Ruyter et al. (Eds.): AmI 2010, LNCS 6439, pp. 197–206, 2010.

A user is requested to read or touch some form of tag (barcode, RFID, IR) embedded into a desired resource. The tag is used to obtain a handle to a digital representation of the resource. Each of the two approaches has its own advantages and disadvantages:

- **Direct tag-based discovery:** The user can directly interpret his/her intention into something that the AmI system can immediately recognize. There is no guessing about what resources the user needs. Moreover, observing resources in the ambient can initiate serendipitous and spontaneous interaction with resources [6]. On the other hand there is little automation in the discovery process. Interaction design becomes a major issue. Another disadvantage is that discovering resources that are not in the line of sight (e.g. a printer in the neighboring room) is not possible.
- **Indirect network-based discovery:** The main advantage is that resources can be discovered and integrated regardless of their location. The disadvantage, or merely the challenge, is to know what resources the user really needs. The AmI system has to guess what resources, among all that are registered, are most relevant to the user in a given, highly dynamic context.

We believe both of these approaches are important to an AmI system, and that the underlying discovery mechanisms should equally support both. But the two approaches are often developed in parallel with little integration. In this paper we will present a system called UbiDisco (Ubiquitous Discovery system) that integrates the two approaches in an open and extensible architecture. In Section 2 we will introduce a scenario and show how an example application based on UbiDisco solves some of the challenges of discovery in AmI. In Section 3 we introduce the architecture of UbiDisco, and in Section 4 we will describe the current implementation of UbiDisco and in Section 5 how it is evaluated so far. Section 6 represents some related work. Section 6 concludes the paper with a discussion of the results so far.

2 UbiDisco Scenario and Main Concepts

This section gives an overview of the main concepts of UbiDisco. UbiDisco is part of the UbiCollab platform [7]. UbiCollab is an experimental platform that aims to support mobile users in remote collaboration. UbiCollab implements the concept of a *human grid*, i.e. a collection of physically distributed people who share a context for collaboration. Participants in a human grid use locally available resources in order to facilitate remote collaboration. Resources play a central role in UbiCollab, and UbiDisco is the subsystem of UbiCollab in charge of discovering ambient and web resources. *Resources* in UbiDisco can be internet-enabled devices, web services, or physical objects with a digital representation. Once a resource is discovered using UbiDisco and integrated into the UbiCollab platform, the resource provides a set of *Services* to UbiCollab *Applications*. Resources are integrated into UbiCollab using *Proxy Services* which are similar to device drivers.

Discovery and integration of resources is done using a set of UbiDisco *Discovery Actions*. A Discovery Action is responsible for guiding the user through a set of interactions in order to obtain a handle to a desired resource. A Discovery Action can be implemented to acquire the handle directly from the resource (using e.g. a tag) or to look up the user's query in an online catalog. The Discovery Action framework of UbiDisco enables a seamless combination of direct tag-based discovery and indirect network-based discovery. The interaction mechanisms implemented by UbiDisco provide a seamless integration of the two also at the user interface level.

2.1 Scenario

This section describes a typical scenario for UbiDisco.

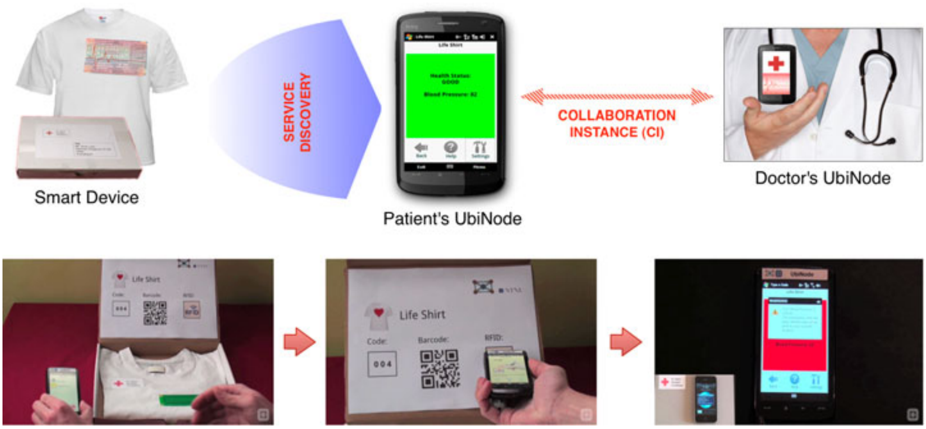

Fig. 1. Scenario illustrating the discovery of lifeshirt by reading an RFID tag

Arne has a chronic heart condition. He is equipped with a UbiNode from his hospital. It is a smart touch-based device that he uses in order to keep in touch with the hospital. The UbiNode has a number of applications installed. One is where he registers his physical condition for his doctor to monitor. During his latest visit to the hospital he is prescribed a lifeshirt, a new intelligent shirt with embedded sensors. After unpacking the shirt, he finds a sheet of paper in the box with instructions about how to use the shirt. It is written that he should add the shirt to his UbiNode. He can choose among three ways of adding the device: 1) take a photo of a barcode on the paper, 2) touch an RFID tag on the paper, 3) punch in a number written on the paper. He clicks on "Add Service" on his UbiNode. Then he selects "Touch RFID" action, and points his UbiNode to the sheet of paper. His UbiNode presents a number of screens with some information and a confirmation page. Then a message shows up telling him that the device was added successfully. The shirt now communicates with his UbiNode using Bluetooth. Arne then starts his medical application. The application notifies him that a lifeshirt is detected and asks him whether he wants to use it.

Some days later, Arne is on a regular visit to his doctor. In the hospital he decides to print the latest measurements he has done. He cannot find a printer service on his UbiNode. He clicks "Add Service" on his UbiNode because he knows he can use it for finding printers. He inspects the possible actions. All of them require him to have a printer close to him. Since he cannot see a printer around, he cannot use any of the actions on his UbiNode. He looks around and finds a tag on the wall with a label "Read this tag to access hospital services." He knows that the tag will give him new options for discovery actions. He clicks "Add new action" on his UbiNode, and takes a picture of the tag on the wall. After confirming on his screen, he now finds a new action in the list of Add Device actions that reads "Search for Hospital's services". He selects the action, and can browse in a list of services, one of them reading "Print documents". He selects the option and is presented with a printer name, and a link to a map that shows where he and the printer are. He follows the map, finds the printer, and adds it to his UbiNode. Next time he knows where to print his documents.

2.2 Analysis

The main point of the above scenario is to demonstrate that resource discovery in AmI needs to be user-centered, i.e. taking users' situation and needs as the starting point. Service discovery is not an act that can be fully automated. Intelligence should be introduced where there is a clear need (demonstrated by the system finding which printer is closest to Arne). In particular the scenario demonstrates why discovery of resources in AmI should be supported both using direct tag-based and indirect network-based discovery mechanisms. Direct tag- based discovery utilizes the properties of the physical space such as proximity and visibility[8], and eliminates a number of risks for "misfiring" i.e. finding the wrong resources.

Although the above points are well-known issues, there are a number of additional innovations represented by our approach. Using a framework for Discovery Actions, and a uniform user interaction mechanism, UbiDisco abstracts away the differences between finding a resource using direct tag-based and indirect network-based discovery. The user is represented by a list of "Actions" that includes examples of both approaches. The goal is to "Add Service". Some actions involve physical activity (e.g. taking a photo or touching an RFID tag) while others involve a mental activity (e.g. browsing a list of resources). The user goes through similar and consistent interaction mechanisms.

3 UbiDisco Architecture

UbiDisco (being part of UbiCollab) employs a peer-to-peer philosophy where all users are equal peers in a human grid, and are represented by a personal device called UbiNode (see Fig. 2). UbiDisco denotes the combination of Resource Discovery Manager and discovery plugins (shown as pieces of puzzle in Fig. 2).

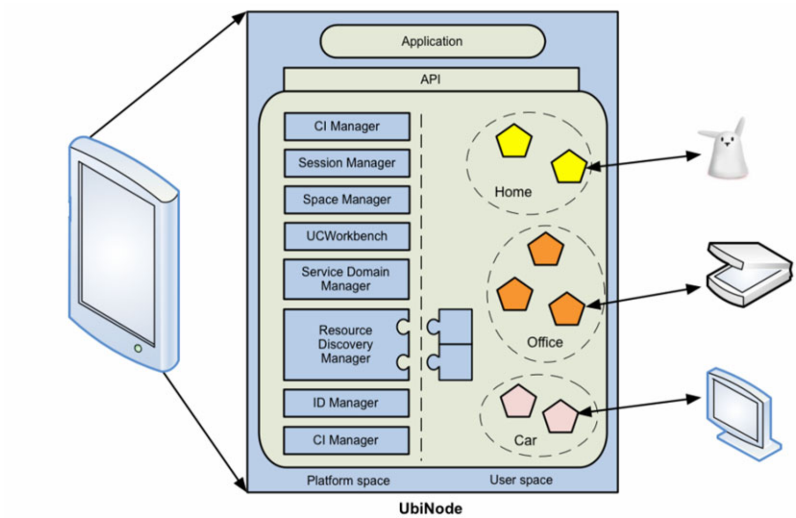

Fig. 2. The architecture of UbiNode

3.1 Overall Service Management Framework

UbiDisco assumes the existence of a proxy layer towards external resources. For each resource that the user wants to use, UbiDisco requires the availability of a Proxy Service. A Proxy Service is code that acts more or less as a traditional driver. Once a Proxy Service is installed on user's UbiNode, UbiCollab applications can use the associated resource. The following steps are performed by UbiCollab service management framework (see Fig.3 on next page):

- **Resource discovery:** Implemented by UbiDisco. During this step a user (or an application) identifies the need for a specific resource. The need is specified in form of a query, and is provided to UbiDisco. A set of discovery plug-ins are notified and start searching for resources. The search can involve various Discovery Actions implemented by each of the plug-ins (Step 1 in Fig.3). Each resource that is discovered is stored in a resource pool along with the ID of the application that asked for the resource.
- **Proxy Service installation:** Implemented by UbiCollab's Service Domain Manager (SDM). The URI obtained during the resource discovery phase by UbiDisco points to an XML document that describes the resource and provides a link to download a Proxy Service for the resource. The URI is passed to SDM (Step 2 in Fig.3). SDM downloads the Proxy Service (Step 3 in Fig.3) into user's Service Domain on the UbiNode (Step 4 in Fig.3). Once the Proxy Service is installed it provides a service interface that can be accessed by applications (Step 5 in Fig.3).
- **Sharing of services:** As an optional step, the user can choose to share a Service Proxy with other users. In this way these users can get access to the local resource. Sharing of resources is a central part of the UbiCollab platform, but is out of scope for this paper and will not be discussed further. Please refer to [7].

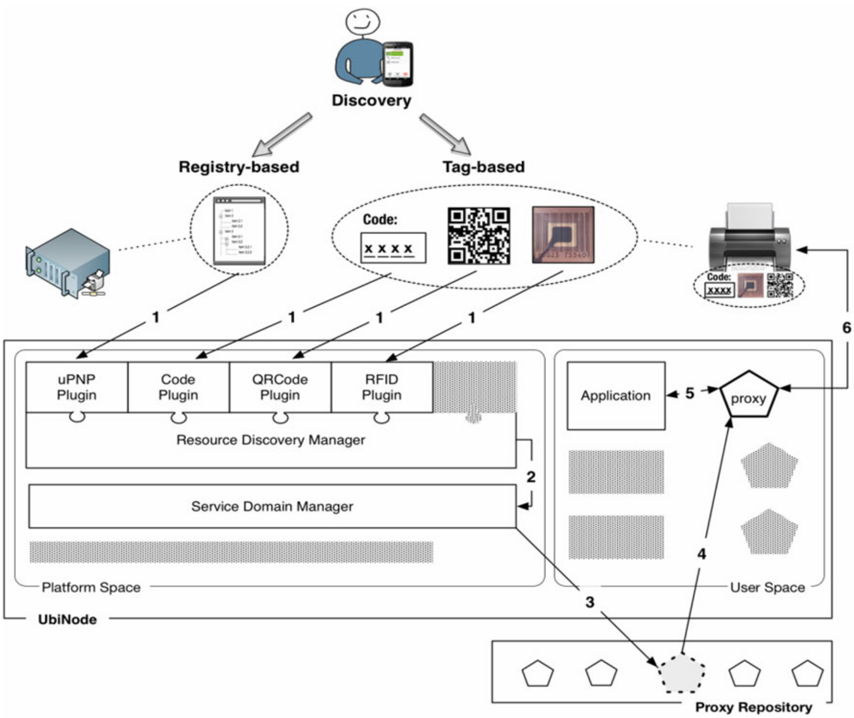

Fig. 3. Overall discovery process in UbiDisco

3.2 Discovery Plug-ins and Discovery Actions

As shown in Fig. 3, the resource discovery phase of UbiDisco is implemented by a set of what we call discovery plug-ins. Plug-ins are developed by third parties and adhere to the framework defined by the UbiDisco. This framework mainly requires an API to allow clients send requests for resources to be discovered, and the implementation of a Discovery Action user interface. Each plug-in that is installed on user's UbiNode will attempt to find the resource according to its own Discovery Action and discovery protocols. The returned value is a set of handles to discovered resources. A plug-in module consists of two parts: a network communication part (implementing the specific discovery protocol such as RFID-based or UPnP), and a Discovery Action part.

At the core of UbiDisco are the various Discovery Actions (DAs) that illustrate innovative ways of user-centered discovery in AmI. A DA is the physical or mental act the user needs to perform in order to be able to start using a newly encountered resource. Various types of DAs can be observed in many existing AmI systems but a coherent framework is lacking. In UbiDisco, the implementation of a DA is done by the discovery plug-ins in a uniform way controlled by SDM. This provides a platform for experimenting with various actions and discovery mechanisms. The following actions have been currently developed for UbiDisco:

- **Touching RFID tags:** The user starts an application that needs a printer. The user observes a printer with an RFID tag. The RFID action is selected by the user, and the user touches the tag with his/her UbiNode. The printer is installed and used by the application.
- **Type a number:** Same as the above, but the action shows a keyboard asking the user to type the number of the printer. The user observes a number tag on the printer and types in the number.
- **Take photo of 2D barcode:** Same as above, but this time the user uses UbiNode's camera to take a photo of the barcode attached to the printer.
- **UPnP:** The user is represented with a list of UPnP services in the network and is asked to select the exact resource.

4 Implementation

UbiCollab platform components, including UbiDisco, are all components that share the same internal architecture. These components are encapsulated into OSGi bundles[1]. User components (applications, proxy service and discovery plug-ins) are also OSGi bundles that are installed via a Discovery Action, and run without rebooting the JVM due to dynamic deployment features of OSGi.

A UbiCollab component contains two layers: the *Platform Abstraction Layer (PAL)* and the *User Abstraction Layer (UAL)*. The PAL contains the basic logic and connects the component to other components and to the OSGi framework. The UAL connects the module to a *User Interface Manager (UIM)*, which in the current implementation is represented by the UCWorkbench; a module built on a eWorkbench draft provided by the Eclipse Foundation. In discovery plugins, PAL implements network communication while UAL implements the discovery action.

UIM is the central user interaction module in UbiDisco (and in UbiCollab). It is the glue between user actions and the underlying component model (see Fig.4). It provides a plug mechanism that allows third-parties to develop Discovery Actions without extensive coding.

A *View* in UbiCollab architecture implements a unit of interaction corresponding to a user task (lowest layer in Fig.4). Similar Views can be grouped into *Perspectives*, which roughly correspond to user intentions (middle layer in Fig.4). For instance, if user's intention is to discover a resource, the user will switch to the Discovery Perspective of UbiCollab, implemented by UbiDisco. There are a number of perspectives in UbiCollab that correspond to different perceived user intentions.

Discovery plug-ins implementing the Discovery Actions have been designed exploiting well-known low-cost object tagging standards as RFID and QRCode tags. The RFID plug-in makes use of high-frequency tags (13.56 MHz), suitable to be scanned by portable readers. Resources URIs are stored in a registry embedded in each discovery plug-in, to allow *off-line* discovery.

[1] OSGi is a Java-based container technology. See www.osgi.org.

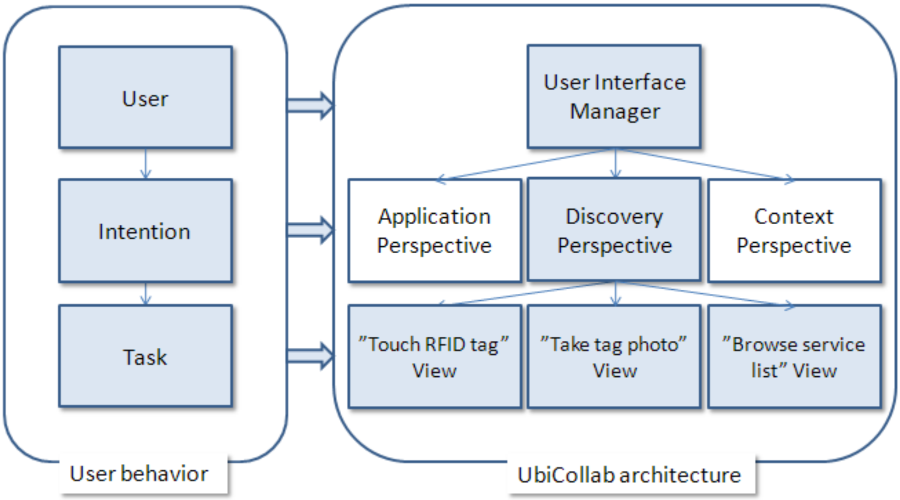

Fig. 4. User Interface Manager manages perspectives and views related to user intention

5 Preliminary Evaluation Results

We are in the early stages of evaluating our results on real users. UbiDisco and the implemented discovery actions have been subject to focus group and hands-on testing by end users of primarily technical background. The groups have varied from 5 to 14, and feedback is collected using Technology Acceptance Model. Preliminary results show potential for improvements in the usability area. The results from these studies have indicated that the relationship between applications and resources is an area that needs more research. The transition from usage perspective to discovery perspective has shown to be confusing to our users. Discovery actions are however considered useful, and the most popular one among our users has so far been the RFID touch action. A more scientific analysis of user feedback will be the subject of a future paper.

6 Related Work

Discovering and integrating resources with users' applications as users move around introduces a number of challenges that are addressed by researchers in the recent years. There are a number of classification schemes documented in the literature that demonstrate a great variety in how resources are represented, how registries are constructed, and how look-up criteria are decided in indirect network-based discovery [3][9]. From being purely network-centred, indirect network-based discovery has become increasingly user-centred. Examples include selection of services based on user activity[10], and selection based on user preferences[11] or context elements such as location and spatial relations[12]. There exist also a number of commercial systems that implement community-based service registries, such as AppStore, Android Market and various UDDI-based directories.

Direct tag-based discovery was pioneered among others by the work on CoolTown[4] and has been developed into its own field of research with an increasing number of interesting results. Augmentation of everyday objects with discovery mechanisms is demonstrated in [5][13]. In [14] an interesting approach is demonstrated where multiple tag-based discovery actions are combined to implement associations among devices. Our research builds on this existing research and extends it with pluggable architectures that allow integration of the two approaches.

7 Conclusions and Future Work

In AmI environments saturated with computational resources, discovery and integration of resources becomes a major undertaking for the users. In professional environments, such as offices, the knowledge at least exists among the users about how to configure and connect resources to the AmI system (but the time to do so is often lacking). In other scenarios, such as ambient assisted living, we cannot assume enough knowledge or capabilities exist among the elderly. In all cases, there is a great potential to automate or simplify a lot of the routine work involved in finding and connecting to the right resources. We have demonstrated how various discovery actions can be integrated into UbiDisco architecture in order to adapt discovery to capabilities and preferences of users and make resource discovery more user-centered.

We are in the process of experimenting with more innovative discovery actions, such as voice-based and the usage of pointing devices. Security and privacy issues haven't yet been considered, but our future plans in this area include digital signing of proxies and plug-ins, and the usage of virtual identities.

References

1. Streitz, N.A., Rocker, C., Prante, T., van Alphen, D., Stenzel, R., Magerkurth, C.: Designing smart artifacts for smart environments. Computer 38, 41–49 (2005)
2. Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Personal Communications 8, 10–17 (2001)
3. Edwards, W.: Discovery Systems in Ubiquitous Computing. IEEE Pervasive Computing 5, 70–77 (2006)
4. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., Spasojevic, M.: People, Places, Things: Web Presence for the Real World. Mobile Networks and Applications 7, 365–376 (2002)
5. Riekki, J., Salminen, T., Alakarppa, I.: Requesting Pervasive Services by Touching RFID Tags. IEEE Pervasive Computing 5, 40–46 (2006)
6. Newman, M.W., Sedivy, J.Z., Neuwirth, C.M., Edwards, W.K., Hong, J.I., Izadi, S., Marcelo, K., Smith, T.F.: Designing for serendipity: Supporting end-user configuration of ubiquitous computing environments. In: Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, pp. 147–156. ACM, London (2002)

7. Farshchian, B.A., Divitini, M.: Collaboration Support for Mobile Users in Ubiquitous Environments. In: Handbook of Ambient Intelligence and Smart Environments, pp. 173–199 (2010)
8. Harrison, S., Dourish, P.: Re-place-ing space: the roles of place and space in collaborative systems. In: Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, pp. 67–76. ACM, Boston (1996)
9. Zhu, F., Mutka, M.W., Ni, L.M.: Service discovery in pervasive computing environments. IEEE Pervasive Computing 4, 81–90 (2005)
10. Bucur, D., Bardram, J.: Resource Discovery in Activity-Based Sensor Networks. Mobile Networks and Applications 12, 129–142 (2007)
11. Park, K., Yoon, U.H., Kim, S.: Personalized Service Discovery in Ubiquitous Computing Environments. IEEE Pervasive Computing 8, 58–65 (2009)
12. Gellersen, H., Fischer, C., Guinard, D., Gostner, R., Kortuem, G., Kray, C., Rukzio, E., Streng, S.: Supporting device discovery and spontaneous interaction with spatial references. Personal and Ubiquitous Computing 13, 255–264 (2009)
13. O'Neill, E., Thompson, P., Garzonis, S., Warr, A.: Reach Out and Touch: Using NFC and 2D Barcodes for Service Discovery and Interaction with Mobile Devices. Pervasive Computing, 19–36 (2007)
14. Broll, G., Haarländer, M., Paolucci, M., Wagner, M., Rukzio, E., Schmidt, A.: Collect&Drop: A Technique for Multi-Tag Interaction with Real World Objects and Information. Ambient Intelligence, 175–191 (2008)
15. Berger, S., McFaddin, S., Binding, C., Hoertnagl, C., Ranganathan, A.: Towards pluggable discovery frameworks for mobile and pervasive applications. In: Proceedings of IEEE International Conference on Mobile Data Management, 2004, Berkeley, CA, USA, pp. 308–319 (2004)
16. ASTRA, `http://www.astra-project.net/`