SCOUT: A Framework for Personalized Context-Aware Mobile Applications

William Van Woensel[1], Sven Casteleyn[1,2], Olga De Troyer[1,2]

[1] Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium
{William.Van.Woensel, Sven.Casteleyn, Olga.DeTroyer}@vub.ac.be

Abstract. The recent evolution in mobile devices, combined with rapid advancements in identification techniques, has lead to new opportunities for mobile application developers: mobile applications that can be made aware of their environment and the objects in it. Furthermore, by combining mobile devices and identification technology with the Web, mobile applications can be developed that exploit interesting services and information associated with nearby objects. We present an application framework that supports the development of such mobile applications, enabling them to become fully context-aware and capable of providing personalized information and services. Because of the decentralized nature of our approach, we allow for more flexibility and scalability, and significantly lower the threshold for third parties to benefit from our approach. Also, due to the separation of concerns and the systematic use of abstraction mechanisms, the framework supports a variety of implementation options (i.e. different identification techniques, push- or pull-based notification mechanisms, etc), while still hiding technical details from the application developer.

Keywords: mobile Web, development framework, context-awareness, mobility, personalization

1 Introduction

In the current mobile Web, users typically use their mobile device (e.g., smart phone, PDA, portable game console) to access the Web using a dedicated mobile browser (e.g., Skyfire, Opera Mini). Although this makes the Web accessible anywhere and anytime, the limitations of mobile devices (e.g., small screen, limited input capabilities, processing power and bandwidth), together with the fact that most Web applications do not take these limitations into account, still hinder the widespread mobile use of the Web. Furthermore, in a mobile setting (e.g., driving, walking, sightseeing), users are often unable or reluctant to spend large amounts of time locating information or services they need at that particular moment and place.

[2] PhD supervisors

Consequently, it is clear that the mobile user experience can be greatly improved by providing the user with personalized information and services. To enable such personalization, the user's full context can be taken into account, comprising personal information (e.g., profile, preferences, needs, ..) and environment information made available by e.g. location technologies (e.g., GPS, triangulation) and identification techniques (e.g. RFID, NFC). However, existing approaches that exploit contextual information suffer from significant problems and drawbacks (see section 2), mostly related to the fact that they employ a centralized system to manage location-specific data and/or context information, which makes them less flexible and scalable.

The goal of this PhD dissertation is to present a decentralized approach to context-specific information/services retrieval that tackles these problems. To that end, in analogy to the CoolTown approach [1-2], we define a mobile environment as a set of distinct physical entities (persons/places/things), some of which have a virtual (Web) presence that provides information or services related to that entity. When the mobile user is nearby an entity, he/she is able to access the entity's Web presence (e.g., by using identification techniques such as RFID, NFC to obtain the Web presence address). The information and services offered are thus location-specific, and can be further personalized by querying the corresponding Web presence for metadata to match with the user's contextual information. This definition makes no assumptions on where information/services are stored, how (the reference to) the Web presence is obtained and how the information/services are personalized.

We present the mobile application development framework SCOUT (Semantic COntext-aware Ubiquitous scouT) that implements these Web presences using Semantic Web technology, while also acting as a platform for custom applications that require context-specific information/services from the user's current environment. Additionally, by allowing the reuse of existing Web resources as Web presences and supporting a gradual adoption of our approach, we significantly lower the threshold for third parties to benefit from our approach.

Using this framework, an application developer can quickly create mobile, context-aware applications while disregarding low-level details concerning context-acquisition. Furthermore, because the different design concerns in the framework are clearly separated, namely entity detection, location management and environment management, we allow different technologies to be used interchangeably.

In the next section, we position our work in the research community. Section 3 provides a global overview of the proposed SCOUT framework. In section 4, we discuss our current status and the challenges facing us.

2 Situating this research

As mentioned in the introduction, our approach employs key concepts first introduced by the HP Cooltown project [1-2]. In Cooltown, Web presences were linked to the corresponding physical entities in order to integrate the physical with the virtual world. The same general concept is applied in [3], where an open lookup infrastructure is proposed in which tagged objects (i.e. containing an RFID tag) are linked to digital information on that object. Many other approaches that aim to provide location-specific information and services [4-6] employ a central information system, which is responsible for storing, maintaining and communicating all location-specific data. In [7-9], a central system is responsible for dealing with context-information associated with a certain region or information system.

Our main contribution is that we focus on the context-specific retrieval of information and services in a decentralized way, while leveraging Semantic Web technology for increased expressive power. This allows for more flexibility and scalability, and also allows users to put online their own Web presences without having to bother with an external, centralized system for content/context management. Furthermore, by automatically providing mobile applications with location-specific information (that can be further tailored using the user's entire context), applications can get the information when (and where) they need it. Our emphasis on context-specific retrieval of information and services is best illustrated by the Environment Model and Filtering Service (see section 3), which allows applications to become aware of the user's surroundings and enables them to become adaptive to changes in the environment, respectively.

3 SCOUT: An overview

The SCOUT framework uses a layered architecture where the different design concerns are clearly separated, assuring independence from underlying technologies. Figure 1 shows an overview of the architecture.

The main responsibility of the first (bottom) layer, the **Detection Layer**, is to provide the upper layers with references to the Web presences of identifiable entities around the mobile user. (The reference is normally provided in the form of a URI.) This layer encapsulates all functionality concerning the actual techniques used to detect entities, as well as the mechanisms to obtain the reference to their Web presence.

The **Location Management Layer** provides the upper layers with a conceptual view on what entities are currently nearby, based on the information provided by the Detection Layer. This is done by creating "positional" relations with nearby entities, and invalidating these relations when the corresponding entities are no longer nearby. For this purpose, specific strategies are employed to determine when an entity is nearby, and when it is no longer nearby.

The third layer, the **Environment Layer**, adds models, functionality and API's that enable the development of context-aware and personalized mobile applications. This layer maintains and provides access to an Entity Model, which stores specific information on the entity (e.g., type, characteristics, preferences, etc). Furthermore, it maintains an Environment Model, which is an abstract representation of the user's current physical environment and is based on the positional relations provided by the previous layer. Applications can pose arbitrarily complex queries to this model: in addition to positional relations, the Entity and Environment Model of related entities can also be referenced in these queries, thus allowing information and services retrieval to be made dependent on the user's entire context. (Clearly, solid privacy policies will need to be put in place when sharing this information!) Finally, in order for applications to become adaptive to the user’s environment, a Filtering Service is provided that allows applications to be alerted when interesting entities become nearby.

The final layer is the **Application Layer**, consisting of the actual applications. Applications that are developed using the SCOUT framework can benefit from all of the functionality described in the previous layers. They may thus personalize information/services according to the user, his/her context and/or location, by using the API’s offered by the framework.

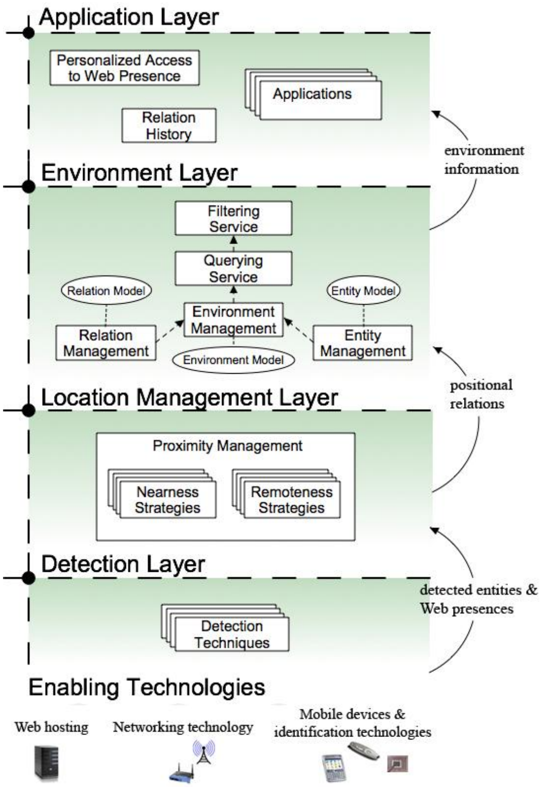

Fig. 1. SCOUT architecture layers.

As mentioned in the introduction, the goal of the SCOUT framework is twofold: on the one hand, it acts as a platform for context-aware, mobile applications; on the other hand, it implements Web presences by allowing external parties (applications) access to the aforementioned models in the Environment Layer. However, the latter is not an absolute requirement: our approach also allows the reuse of existing Web resources as Web presences. For instance, the simplest way of creating a Web presence would be to make the reference to an existing Website available nearby the associated entity. However, it is clear that only the user's location can be taken into account when retrieving such Web presences, because no additional metadata can be queried. In case machine-readable (RDF) metadata is (also) available, further personalization of the location-specific information/services can occur. In the most advanced case, the framework is fully supported by the Web presence, and any information present in the Environment Layer of that Web presence can be queried (if allowed) by external applications for more extensive, powerful personalization of information and functionality.

4 Current status and challenges

As our framework represents a bottom-up approach, our work has been focused on implementing the bottom two layers. Currently, we are working on a (preliminary) version of the Environment Layer. The next step is to implement and test (in a real-life environment) a number of proof-of-concept applications on top of the framework, not only fix possible problems and refine the framework but to also illustrate its usefulness for both users and application developers. Subsequently, we aim to extend the existing Environment Layer to increase security and efficiency. Among other things, this means that strategies will have to be worked out for the efficient and scalable querying of the semantic information of nearby entities, and that solid authorization and privacy policies will have to be put in place.

References

1. Debaty, P., Caswell, D.: Uniform Web Presence Architecture for People, Places, and Things. In: Personal Communications, Issue 4, Volume 8, pp. 46--51, IEEE (2001).
2. Debaty. P., Goddi, P., Vorbau, A.: Integrating the Physical World with the Web to Enable Context-Enhanced Services. Technical report, Hewlett-Packard (2003).
3. Roduner, C., Langheinrich, M.: Publishing and Discovering Information and Services for Tagged Products. In: 19th International Conference on Advanced Information Systems Engineering, pp.501--515, Springer Berlin / Heidelberg, Trondheim, Norway (2007).
4. Tummala, H., Jones, J.: Developing Spatially-Aware Content Management Systems for Dynamic, Location-Specific Information in Mobile Environments. In: 3rd ACM international workshop on Wireless mobile applications and services on WLAN hotspots, Mobility support and location awareness, pp. 14--22, ACM, Cologne, Germany (2005).
5. López-de-Ipiña, D., Vazquez, J.I., Abaitua, J.: A Context-aware Mobile Mash-up Platform For Ubiquitous Web. In: 3rd IET International Conference on Intelligent Environments, pp. 116--123, IEEE, Ulm, Germany (2007).
6. Challiol, C., Rossi, G., Gordillo, S., De Cristófolo, V.: Designing and Implementing Physical Hypermedia applications. In: ICCSA 2006, UWSI 2006, pp. 148--157, Springer Berlin / Heidelberg (2006).
7. Cappiello, C., Comuzzi, M., Mussi, E., Pernici, B.: Context Management for Adaptive Information Systems. Electronic Notes in Theoretical Computer Science 146, 69--84 (2006).
8. Xue, W., Pung, H., Palmes, P.P., Gu, T.: Schema matching for context-aware computing. In: 10th international conference on Ubiquitous computing, pp. 292--301, ACM, Seoul, Korea (2008).
9. Judd, G., Steenkiste, P.: Providing Contextual Information to Pervasive Computing Applications. In: 1st IEEE International Conference on Pervasive Computing and Communications, pp. 133--142, IEEE Computer Society, Fort Worth, Texas, USA (2003).