Involving tourism domain experts in the development of context-aware mobile services

David Martín[a], Diego López de Ipiña[b], Carlos Lamsfus[a] and Aurkene Alzua[a]
[a]Human Mobility and Technology
CICtourGUNE – Competence Research Centre in Tourism, Spain
davidmartin@tourgune.org, carloslamsfus@tourgune.org, aurkenealzua@tourgune.org

[b]Deusto Institute of Technology - DeustoTech
University of Deusto, Spain
dipina@deusto.es

Abstract

Tourists request information and consume services anytime and anywhere by the use of their mobile phones. This information has to be personalized according to their situation in order to better satisfy their needs. However, the development of mobile tourism services that use context data to customize information is complex. One of the main tasks in the development process is the definition of high level situations. These situations are best defined by people with expertise in the tourism domain that usually do not have technical skills. With the existing context-aware development toolkits, the involvement of non-technical domain experts is quite complex. This article presents an innovative Cloud-based platform to ease the development of context-aware mobile services that provides several wizards in order to manage context data by non-technical users.

Keywords: Tourism Guide, Context-Aware, Mobile Service, Cloud Computing, End-user Development.

1 Introduction

The personalization of information and services is essential for an optimal user experience. Furthermore, this customization becomes even more important in mobile scenarios, where users have to interact with small devices and they usually require very specific information at a given time (Grün, Werthner, Pröll, Retschitzegger & Schwinger, 2008). This way, it is crucial for mobile services to be able to acquire data about the user's context on the move, process it in order to identify the user's situation at a given time and place and finally adapt the available functionality to that situation (Dey, Abowd & Salber, 2001).

Several context-aware programming toolkits have been proposed in order to simplify the development of context-aware services (Baldauf, Dustdar, & Rosenberg, 2007). However, there are some gaps in the reviewed frameworks that make their application difficult to the tourism domain. On the one hand, not all of them are designed to support users' mobility, and do not provide GIS functionalities in order to perform geospatial operations. On the other hand, the reviewed frameworks offer high-level programming APIs for skilled programmers. This makes the involvement of non-

technical stakeholders in the development life cycle almost impossible. The participation of users that do not have programming skills but are experts in the tourism domain can speed up and improve the development process of context-aware mobile tourism services.

This article presents an innovative solution, called Context Cloud, which aims to address the identified gaps in terms of end-user programmability of earlier Context-Aware development toolkits. Context Cloud is an infrastructure with a web front-end where all the aspects involved in the development of context-aware mobile services can be easily configured.

The paper is organized as follows. In section 2, the architecture of the platform is described. Section 3 contains the description of the main functionalities of Context Cloud. In section 4, the related work is discussed. Section 5 concludes the paper with brief concluding remarks.

2 Description of the Architecture

Context Cloud can be considered as a black box that receives inputs from the identified context sources and produces outputs triggered by the user defined ECA rules (Beer, Höpken, Rasinger, Fuchs & Werthner, 2007). The interactions between the platform and the third party services (e.g. context sources, web services, mobile devices) are made in a RESTful way, following the best practices established by the community of the Web of Things (Guinard, Trifa, Mattern & Wilde, 2011).

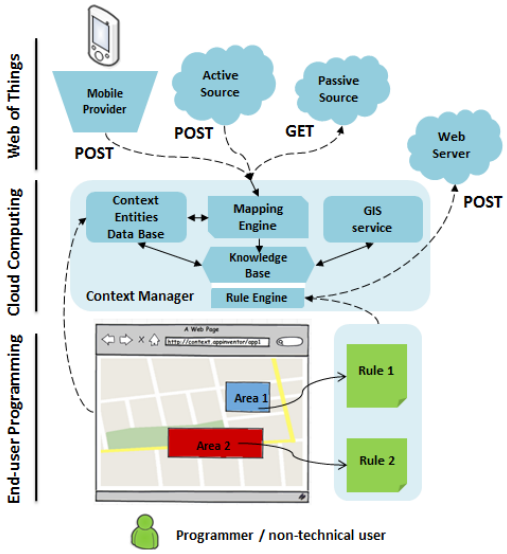

Fig. 1. Context Cloud Architecture

Figure 1 shows the system architecture, which is divided into three different layers. The first layer represents the web front-end, where the user can create areas, rules,

context entities, mappings between context sources and entities and the accesses to the identified context sources by the use of wizards.

The middle layer contains the Context Manager. This component manages the context data life cycle. Here, the defined context data model is stored in the Context Entities Data Base. Rules are also validated and inserted into the Rule Engine. The Mapping Engine does the mappings between data coming from the configured context sources and the defined context entities, according to the user's defined mappings. It also saves or updates context entities instances into the Knowledge Base. The GIS Service translates the coordinates of context entities into a registered area name. This way, rules will only have into account context entities that are in their associated area. Finally, the Rule Engine is the responsible of firing all defined rules. This component can also POST data to external web services as a consequence of the defined rules.

The upper layer contains all the context sources. These sources can actively POST XML data to the platform or they can be accessed by the platform itself using periodical GET requests.

3 Description of the Functionalities of Context Cloud

Context Cloud has different features that are configurable using a web interface which is divided into five different portlets that show information about the configured items: areas, rules, passive sources, active sources and context entities.

First, the platform allows the creation of a context model defining entities with different properties (e.g. Integer, String, Boolean). The platform transforms the defined entities into Java Bean type classes in order to make them manageable by the proposed knowledge base. In addition, context data acquisition from external context sources can be configured. For that purpose, Context Cloud provides two types of wizards. The first one can be used to gather data from passive sources, i.e., the platform can be configured to make periodic GET requests to the selected web service and get an XML document with the data in response. The second wizard allows the creation of active sources which make POST requests containing XML context data. The obtained data is automatically processed by the platform and made it available to the web front-end in order to configure the mappings between the received data and the created context entities.

The platform also provides wizards in order to create areas and rules assigned to these areas, which are defined by polygons created over a Google Maps layer. Rules are configured using the syntax provided by the Drools Expert System[1], which has been used to implement the knowledge base. Rules can be edited using a wizard that translates user's settings into Drools syntax automatically. The consequence of each rule can trigger a POST request to any external web service end-point in order to send information that is related to the triggered rule.

[1] http://www.jboss.org/drools

Therefore, thanks to the designed platform, users without any programming skills can actively participate in the development of context-aware services. In this development life cycle, the creation of the context model, the definition of areas, the creation of rules assigned to each of these areas and the mappings between context entities and sources can be configured by non-technical users, who are experts in the application domain. This way, programmers can focus on the interactions between the platform and the context sources and on managing all the outputs that are triggered by the rules in order to develop all the business logic of the services to be implemented.

4 Related Work

Several toolkits have been proposed in order to support the development and deployment of contextual computing applications. The CASS (Fahy & Clarke, 2002) framework is a layered middleware architecture that uses a relational data model to represent context data. JCAF (Bardram, 2005) is a framework and a runtime environment to develop and deploy contextual computing applications. The CoBra middleware (Chen, Finin, Kagal, & Perich, 2004) proposes a different approach where software agents are used in order to acquire and process context data in a smart meeting room environment. SOCAM (Gu, Pung, & Zhang, 2005) is based on three different layers, namely a sensing layer, a middleware layer and an application layer. Recent projects, like DiaSuite (Cassou, Bruneau, & Consel, 2010), propose new visual approaches where domain expert users can be involved in the development life cycle.

Mobile frameworks have also been developed in order to create applications that are executed in mobile devices (Hofer, Schwinger, Pichler, Leonhartsberger, & Altmann, 2002). The main drawback of mobile frameworks is that they are not powerful enough to support complex context management and reasoning.

Context Cloud covers the gaps found in the reviewed frameworks, by providing additional functionalities. For instance, it is the only toolkit that supports an automatic context data life cycle management. This includes the automatic conversion from gathered raw context data to the context data model and automatic updates of current context data and storage of past context data. It also deletes context entities instances from the knowledge base when a registered context source is unregistered or it is no longer available. A geographic information system is included in order to support user's mobility. Context Cloud is the only toolkit that has a web front-end, which makes it accessible from any Internet connected device and it provides wizards in order to configure the needed functionalities, without having to code any programming line. This way, Context Cloud represents a more comprehensive approach to the development of context-aware mobile tourism services. The new platform supports the active participation of both tourism experts and developers. Developers may just concentrate on technical functionalities, while tourism domain experts may focus on modelling tourists' needs as well as on configuring the platform's behaviour.

5 Conclusions

This article presents an innovative way of implementing context-aware mobile services. The platform eases the development of context-aware applications, influenced by the Web of Things, Cloud Computing and End-user programming paradigms.

Usually programmers do not know anything about the tourism domain and it can be difficult for them to identify the needs of tourists on the move, even with the help of tourism experts. The aim of this platform is to involve tourism domain experts with no programming skills in the development process of these kinds of services. Like that, the desired functionalities are best modelled and the final product can be better adapted to the tourism market.

References

Baldauf, M., Dustdar, S. and Rosenberg, F. (2007): A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, p. 263.

Bardram, J. E. (2005). The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications. In H. Gellersen, R. Want, & A. Schmidt, Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005) (Vol. 3468, pp. 98-115). Munich, Germany: Springer Verlag.

Beer, T., Höpken, W., Rasinger, R., Fuchs, M., Werthner, H. (2007). Exploiting E-C-A Rules for Defining and Processing Context-Aware Push Messages. In: *Proc. of International RuleML Symposium on Rule Interchange and Applications (RuleML)*, Orlando, Florida, pp. 199-206.

Cassou, D., Bruneau, J., & Consel, C. (2010). A tool suite to prototype pervasive computing applications. 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 820-822.

Chen, H., Finin, T., Kagal, L. and Perich, F. (2004). Intelligent agents meet the semantic Web in smart spaces. IEEE Internet Computing, vol. 8, pp. 69-79.

Dey, A., Abowd, G., & Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction, 16(2), 97-166. doi: 10.1207/S15327051HCI16234_02.

Fahy, P., & Clarke, S. (2002). CASS - Middleware for Mobile Context-Aware Applications. Ieee Pervasive Computing.

Grün, C., Werthner, H., Pröll, B., Retschitzegger, W., & Schwinger, W. (2008). Assisting Tourists on the Move- An Evaluation of Mobile Tourist Guides. 7th International Conference on Mobile Business, 171-180. Ieee. doi: 10.1109/ICMB.2008.28.

Gu, T., Pung, H., & Zhang, D. (2005). A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications, 28(1), 1-18. doi: 10.1016/j.jnca.2004.06.002.

Guinard, D., Trifa, V., Mattern, F., Wilde, E. (2011). From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices. In: Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.): Architecting the Internet of Things. Springer, ISBN 978-3-642-19156-5, pp. 97-129.

Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., & Altmann, J. (2002). Context-awareness on mobile devices - the hydrogen approach. Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 292-302.