MANAGING CONTEXT DATA IN PERSONAL SMART SPACES

Nicolas Liampotis, Nikos Kalatzis, Ioanna Roussaki, Efstathios Sykas and Miltiades Anagnostou
School of Electrical & Computer Engineering, National Technical University of Athens, Athens, Greece

Keywords: Personal Smart Space, Context management, History of context, Context refinement, Community-awareness.

Abstract: Current research on pervasive computing is opening the way for convergence between mobile telecommunications and the Future Internet. This paper introduces a novel approach to this convergence in the form of the self-improving Personal Smart Spaces (PSSs). One of the core aspects that need to be supported within PSSs is the management of the user related context information. This paper elaborates on the mechanisms that have been designed and implemented in order to address the advanced requirements of PSSs regarding the establishment of a robust distributed context management framework.

1 INTRODUCTION

Pervasive computing (Schmidt, Spiekermann, Gershman, and Michahelles, 2006) is a next generation system paradigm that aims to assist users in their everyday tasks in a seamless unobtrusive manner. It assumes that users are surrounded by numerous communication and computing devices of various features, which interoperate and are capable of capturing and processing information regarding users, their behaviour and their environments. In this framework, there have been various research initiatives aiming towards the design and realisation of smart spaces (Singh, Bhargava and Kain, 2006), where various automation facilities support the users. However, these are fixed spaces that provide pervasive features and adapt to the user needs in a static and geographically limited environment.

Nevertheless, when dealing with mobile users, different and more challenging problems are introduced and need to be resolved (Hansmann, Merk, Nicklous and Stober, 2003; Huang and Mangs, 2008). In this case, the users require the same pervasive services wherever they are and whatever devices they carry along.

Personal Smart Spaces (PSSs) (Roussaki et al., 2009) aim to couple the facilities offered by next generation mobile communications with the features provided by the static smart spaces to support a more ubiquitous and personalised smart space that is able to follow the user wherever he/she goes. A PSS provides to its owner multimodal intelligent interfaces, via which he/she is able to access and configure the various services and resources that are available locally and remotely, even when limited or even no network connectivity is available. PSSs are able to discover other PSSs and interact with them in order to create a richer and more flexible environment for their owners. Each PSS consists of multiple devices, both mobile and fixed, owned by a single user. PSSs constantly monitor their owner's behaviour & environment and they exploit learning techniques to further optimise the pervasive experience perceived by their owners.

In PSSs, network operators, individual sensors, sensor networks or even web and computing resource managers capture valuable information (e.g. device location/status, user profiles, movement patterns, user activities, network performance) and exploit them to provide enhanced services to users. These are called context-aware services (Anhalt et al., 2001; Loke, 2006) and are based on the exploitation of context data such as the above.

The PSS Context Management (CM) framework acts as an intermediate layer between PSS/3rd party context-aware services and the sources of context information. The components comprising the CM architecture are: Context Broker, Context Refinement, Context History Management, Context DB Management, and Context Source Management. A description for each component is provided in subsequent sections of this paper. Apart from the components above, the CM architecture comprises a Database Management System (DBMS) which enables access to the actual context repositories, i.e. the Context Database and the History of Context

Kalatzis N., Liampotis N., Roussaki I., Sykas E. and Anagnostou M..
MANAGING CONTEXT DATA IN PERSONAL SMART SPACES.
DOI: 10.5220/0003374302880291
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2011), pages 288-291
ISBN: 978-989-8425-48-5

Database (HoC DB). The former database is used for current context information, while the latter stores past data.

This paper briefly presents the functionality offered by the components above. More specifically, it is structured as follows. In Section 2, the features implemented by the components that are responsible for sensing, maintaining, managing and distributing context information are descibed. In Section 3, the historic context data management approach, as well as the context refinement mechanisms are shortly presented. Section 4 elaborates on the requirements that need to be addressed to extend the PSS CM framework so that community-related features are supported. Finally, in Section 5, the paper conclusions are drawn.

2 CONTEXT SENSING, MANAGEMENT AND DISTRIBUTION

In this section, the functionality of the CM components, which are responsible for sensing, managing and distributing context information, is briefly presented.

2.1 Context Broker

Access to context information is managed by the Context Broker. This component provides a query interface for retrieving, updating, adding or removing context data. Context queries can be classified as follows: (i) *CtxIdentifier-based*, where Context consumers can specify context IDs in their context queries in order to identify the data items they are interested in; (ii) *location-based*, which is actually a special use case for ID-based queries and allows for discovering new context information based on location hierarchies modelled by context entities and their associations; (iii) *temporal-based*, which provides context consumers with access to past values of context data. The actual processing of such data is managed by the ContextHistoryMgmt component which is detailed in Subsection 3.1. It should be noted that context retrieval queries can be performed either synchronously or asynchronously.

Apart from context query handling, the Context Broker supports a hybrid distributed-centralised management of context data. The hybrid approach was chosen over a centralised or full distributed one in order to address scalability issues in terms of processing, storage, remote communication and, power consumption. More specifically, in this approach, each PSS device contributes to collecting context information and is assigned with a different role based on its capabilities. Although every PSS member hosts a context database, context information is not replicated in all of them. There is, however, a single device (*master*), typically one with server capabilities, i.e. high processing power and storage capacity, whose database contains every piece of context information and is thus able to handle all context queries. In addition, the master device is responsible for maintaining historic context data in order to support inference of current and future context. Regarding the other (*slave*) devices in the PSS, context queries cannot always he handled locally and must therefore be forwarded to the master device for further processing. If the master device becomes unavailable, a slave one may be elected in replacement.

2.2 Context DB Management

The PSS CM framework maintains two databases for storing context information; one for current values and the other for historical data. The scope of the Context DB Management (CDBM) component is to serve as an intermediate layer between the context query interface provided by the Context Broker and the underlying DBMS that actually controls the storage, management and retrieval of data in these two databases. Therefore, this intermediate layer is responsible for translating context queries submitted by platform and 3rd party context-aware services via the Context Broker component into standard SQL queries that can be executed in the framework's DBMS. More specifically, the CDBM component utilises Hibernate (Hibernate, 2010), an object-relational mapping library, in order to map context model objects, i.e. entities, attributes & associations, to records in a traditional relational database.

2.3 Context Source Management

The Context Source Management (CSM) component serves as an interface between a diversity of context sources and the Context Database. These sources include sensors, as well as, context-providing services that are able to feed the CM framework with context information following a source registration process. Furthermore, the CSM is responsible for monitoring the quality characteristics of available context sources in order to be able to select the most appropriate source in the presence of multiple sources for the same type of context information. This decision is based on the QoC requirements posed by context consumers. Such

requirements may also lead to the re-configuration of registered context sources.

3 CONTEXT HISTORY AND REFINEMENT

The functionality of the CM components that are responsible for managing historic context data, as well as, refining context information is described in the subsections that follow.

3.1 Context History Management

The main functionality of the Context History Management component is to collect and manage History of Context (HoC) data (Kalatzis et al., 2008). Maintaining these data is of great importance to the CM, as it enables the system to provide old data that are no longer stored in the Context DB. This is even more critical for other components of the PSS architecture, as it supports the provision of the necessary training data sets to the components that facilitate the self-improving and pro-active behaviour. As already stated, a PSS consists of various devices, each of which hosts a CM system. However, the HoC database is maintained only on the *master* device of the PSS, which has rich storage and processing resources. It should be pointed out that the CM does not automatically monitor all types of context information and store the respective past values in the HoC database. Instead, the services/components interested and/or the PSS owner or a system administrator have to explicitly specify the context types to be maintained in the HoC database.

There are several components of the PSS framework that make use of the stored historic data. Such a component facilitates predictions of the user's next action regarding the usage of pervasive services. Context history data sets are also used for creating profiles of user's preferences, for providing recommendations regarding service usage and are also used by context reasoning algorithms.

Retrieving historic data for a specific context attribute is a straightforward process provided by various methods of the Context History Management interface. Those methods implement standard SQL queries based on the attribute CtxIdentifier in question and criteria related with the timestamps or/and the values of the records.

However, there are cases where context history consumers require data sets containing sequences of context data escorted by other context values occurred at the same time. For example, a sample data set containing the values for a sequence of a context attribute A escorted by the values of context attributes B and C will form tuples, as follows:
<CtxAttrA=val1, CtxAttrB=val2, CtxAttrC=val3>
<CtxAttrA=val4, CtxAttrB=val5, CtxAttrC=val6>
<CtxAttrA=val1, CtxAttrB=val2, CtxAttrC=val8>
In order for the HoC component to support these types of queries, an additional registration process is required prior to the storage of the records. The Context History Management provides methods allowing the history consumer to specify the primary attribute identifier to be monitored along with the identifiers of the escorting attributes. After this registration process, every update of the primary attribute will trigger the storage of each of the escorting attributes in the HoC DB along with the primary attribute.

3.2 Context Refinement

Context refinement implies the processing of raw context data captured by sensors and other sources, in order to extract high level context information. The overall context refinement process consists of two main steps. The system initially extracts context inference rules based on context history data. This process takes place on the *master* device. The inference rules are then distributed to all PSS devices capable of performing context refinement. When a context consumer submits a request for context information that is neither available by sensors, nor lies in the context database, the refinement process is triggered automatically in an attempt to provide the necessary results. The actual inference takes place based on the distributed rules that are used to process raw sensor data. Thus, even the resource constrained devices are capable of performing context refinement locally without having to contact another system. The overall design of the context refinement mechanisms and the implemented inference algorithms are extensively described in (Frank et al., 2009).

4 EXTENDING THE CM IN SUPPORT OF COMMUNITIES

The described CM approach fulfils all the necessary requirements in order to support context awareness in mobile and fixed PSSs. However, as the focus of PSS is the individual rather than the communities of users, an important part of human behaviour is neglected: socialising. Social computing, on the

other hand, has enjoyed meteoric success in bringing people together online. The time is ripe for these two paradigms to converge. But in order to extend the PSS beyond the individual to dynamic communities of users, the context management system described in the previous sections needs to address additional requirements, thus introducing several research challenges. The new requirements that need to be met include the following:

- Community context modelling, management and inference
- Shared community context and inherited context from hierarchical communities
- Dynamic community creation and membership based on context similarities
- History of community context management and exploitation
- Community enhanced locating of individuals with user involvement in location tagging

All requirements above introduce new promissing research fields that are currently being investigated by the authors.

5 CONCLUSIONS

Personal Smart Spaces (PSSs) is a notion introduced to bridge the gap between isolated fixed smart spaces and eliminate the islands of pervasiveness that may exist, where users have no access to pervasive services while being disconnected from fixed smart spaces. The delivery of pervasive services in PSSs relies on the implementation of a context management system that enables third party and PSS services to utilise context data and interact with the user at the appropriate time and in the appropriate manner and get personalised and adapted to the user's requirements and context. The context management system collects & stores these context data and maintains the history of context information. It also provides context data to other PSS components in support of the learning, reasoning and self-improvement features of PSSs. The CM framework presented in this paper is extendible to address the needs not only of individual users, but also of dynamic communities of users. This extension introduces several research challenges and new concepts, such as community context, community-enhanced context handling and inference, etc. The implementation of the PSS context management system framework started in the beginning of 2009. The first prototype has been delivered in September 2009 and since then various versions were released. The final and complete version of the PSS framework was released on September 2010 (http://psmartspace.sourceforge.net/).

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 215098 of the PERSIST (*PERsonal Self-Improving SmarT spaces*) Collaborative Project, as well as under grant agreement n° 257493 of the SOCIETIES (Self Orchestrating CommunIty ambiEnT IntelligEnce Spaces) Collaborative Project.

REFERENCES

Anhalt, J., Smailagic, A., Siewiorek, D., Gemperle, F., Salber, D., Weber, S., Beck, J., Jennings, J., 2001. Toward Context-Aware Computing: Experiences and Lessons, *IEEE Intelligent Systems*, 16(3), 38-46.

Frank, K., Robertson, P., McBurney, S., Kalatzis, N., Roussaki, I., Marengo, M., 2009. A Hybrid Preference Learning and Context Refinement Architecture, *PERSIST Workshop on Intelligent Pervasive Environments* (AISB 2009), Edinburgh, Scotland.

Hansmann, U., Merk, L., Nicklous, M. S., Stober, T., 2003. *Pervasive Computing: The Mobile World*, New York, USA: Springer Professional Computing.

Hibernate, 2010. Retrieved November 11, 2010, from http://www.hibernate.org/

Huang, S., Mangs, J., 2008. Pervasive Computing: Migrating to Mobile Devices: A Case Study. In *2nd Annual IEEE Systems Conference*. Montreal, Quebec, Canada.

Kalatzis, N., Roussaki, I., Liampotis, N., Strimpakou, M., Pils, C., 2008. User-centric inference based on history of context data in pervasive environments, *3rd ACM International Workshop on Services Integration in Pervasive Environments* (SIPE'08), Sorrento, Italy.

Loke, S. (2006). *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*. Florida: Auerbach Publications.

Roussaki, I., et al. (2009). *Persist Deliverable D2.5: Revised Architecture Design*, Technical report, PERSIST project (FP7-ICT-2007-1).

Schmidt, A., Spiekermann, S., Gershman, A., Michahelles, F. (2006). Real-World Challenges of Pervasive Computing. *IEEE Pervasive Computing*, 5(3), 91-93.

Singh, R., Bhargava, P., Kain, S.(2006). State of the art smart spaces: application models and software infrastructure. *ACM Ubiquity*, 7(37), 2-9.