Context Acquisition, Representation and Employment in Mobile Service Platforms

Anna V. Zhdanova, Josip Zoric, Marco Marengo, Herma van Kranenburg, Niels Snoeck, Michael Sutterer, Christian Räck, Olaf Droegehorn, Stefan Arbanowski

Abstract— **Mobile environments, the Web, services, semantics, converge in the physical world forming a shared communication sphere. To ensure context-enabled interoperation on service platforms in such a sphere, developers need to have a shared specification of objects belonging to the sphere and their roles. We present context acquisition, context representation, context enabling and use in mobile service platforms, outline the main ontological enablers of the shared communication sphere, and illustrate their added value with a scenario.**

***Index Terms*— context awareness, service platforms, ontologies, Internet, WWW, mobile applications**

I. Introduction

THIS paper includes a presentation of ideas on how a mobile service platform, specifically, the platform being designed in the IST project SPICE can utilize and benefit from the context information arriving from heterogeneous context sources (such as physical sensors, ontology-enabled profiles, WWW/Internet, system data, etc.). The motivation of the work is to attain integration of the state of the art *context acquisition and representation* practices on mobile service platforms, making the corresponding components *mutually interactive* within and across service platforms, and *enabling and use* of context in building services and applications.

By *context* we understand any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects [3]. In particular, context comprises a subset of available information that is relevant to a specific event.

By a *service platform* we understand the following:

- an environment for services and applications to operate in, standardized access to the data of system and physical communication layers,
- a provider of service enablers, i.e., components that make creation of other services easier and provide end-users with a transparent service access across heterogeneous networks and domains:
    - specific functionality support: call control, instant messaging, streaming, location retrieval, etc.
    - general service management: service discovery, composition, brokering, mediation, QoS management, AAA/A4C management, etc.
- an aggregator, manager and provider of the context data coming from outside of the system (e.g., from the Web) and physical layers.

This paper is organized as follows. Section 2 focuses on context acquisition, and Section 3 focuses on context representation/formalization. Motivating scenario, context enabling and its use in services and service platform are described in Section 4. Section 5 concludes the paper.

II. Context acquisition

In this section, we analyze context acquisition for service platforms from the perspectives of the physical world, enabling systems and the Internet/WWW systems.

A. Sensors/Physical World

The typical physical world context information acquired from sensors can be classified into seven categories: *movement/acceleration*, *light*, *proximity*, *audio*, *temperature*, *mechanical force*, and *humidity* [1].

Whereas much research on context acquisition from sensors focuses on operating with information related to movement and location, some areas of context acquisition are not sufficiently covered. Such areas include acquisition of information of visual character (primarily due to the complexity of its acquisition), information which is acquired by humans via their senses of taste and smell, and human emotions and mood. Addressing the challenges above, i.e.,

Manuscript received April 16, 2006. This work has been partly supported by European Commission under IP SPICE (IST-2006-027617).

A. V. Zhdanova is with the Centre for Communication Systems Research, University of Surrey, Guildford, GU2 7XH, UK (phone: 44-1483-684745; fax: 44-1483-686011; e-mail: a.zhdanova@surrey.ac.uk).

J. Zoric is with Service Platform Group, Telenor R&D, Trondheim, Norway (e-mail: josip.zoric@telenor.com).

M. Marengo is with Telecom Italia, Torino, Italy (e-mail: marco.marengo@telecomitalia.it).

H. van Kranenburg is with Telematica Instituut, Enschede, The Netherlands (e-mail: Herma.vanKranenburg@telin.nl).

N. Snoeck is with Telematica Instituut, Enschede, The Netherlands (e-mail: Niels.Snoeck@telin.nl).

M. Sutterer is with University of Kassel, Kassel, Germany (email: michael.sutterer@comtec.eecs.uni-kassel.de).

C. Räck is with FOKUS, Fraunhofer, Berlin, Germany (e-mail: christian.raeck@fokus.fraunhofer.de).

O. Droegehorn is with University of Kassel, Kassel, Germany (e-mail: Droegehorn@uni-kassel.de).

S. Arbanowski is with FOKUS, Fraunhofer, Berlin, Germany (e-mail: stefan.arbanowski@fokus.fraunhofer.de).

acquiring and utilizing context information of previously unexplored types has a potential to bring innovative services.

Other general challenges for context acquisition from a physical perspective include addressing distribution, heterogeneity and scalability challenges:

- Sensors are distributed over various (heterogeneous) mobile and fixed devices.
- Sensors are distributed over various administrative domains, e.g., home and office environment.
- In a real world context-aware system, many sensors are deployed, and a growing number of services are expected to utilize the produced sensor information generating an increased information flow.

B. *Enabling Systems Example (CMF)*

The **Context Management Framework (CMF)** is a distributed middleware framework that facilitates the acquisition of heterogeneous contextual information across multiple devices and administrative domains. The framework facilitates lifecycle and information transfer management of context acquisition and processing services, and can be deployed on all JAVA-enabled devices, ranging from smartphones to corporate servers.

Context acquisition services are called *ContextWrappers* within the CMF [9]. Each *ContextWrapper* encapsulates a predefined type of contextual information, such as the location, time, presence, temperature. In a typical deployment scenario, multiple *ContextWrappers* are physically distributed over various devices, and a single device hosts multiple *ContextWrappers*, continuously feeding a wide range of sensor information into the framework. Multiple *ContextWrappers* that provide the same type of information can coexist within the framework, enabling the creation of redundancy mechanisms to improve the robustness of the information acquisition process.

C. *Internet and WWW Systems*

Mobile environments and the Web converge forming a shared communication sphere. This causes appearance of new settings to be supported, e.g., when the user utilizes mobile and fixed devices to interact with systems. Interaction and connectivity of mobile applications with the Internet increase. To ensure interoperation of mobile and Web applications and tools (running on various service platforms in such a sphere), developers need to have a shared specification of objects belonging to the sphere and their roles. Certain ontologies have already been developed for mobile communications area with employment of Semantic Web formalisms [5; 6]. However, widespread and global adoption of such ontologies remains a challenge.

III. Context Representation

In this section we analyze context and user profile representation and management in mobile service platforms. Further, we outline an approach to context representation that can be enabled on the service platforms.

A. *Context Representations across Service Platforms*

Service platforms (SP) are heterogeneous; many of them structurally and functionally different. Nevertheless they converge, and the common services can be executed in the shared communication sphere (containing the elements of various SPs). Context enabled interoperation implies that the functionality and information from various SPs (see Figure 1) can be combined in common services. Prior to collaboration, the information and knowledge should be processed and shared, to ensure the information and service precision.

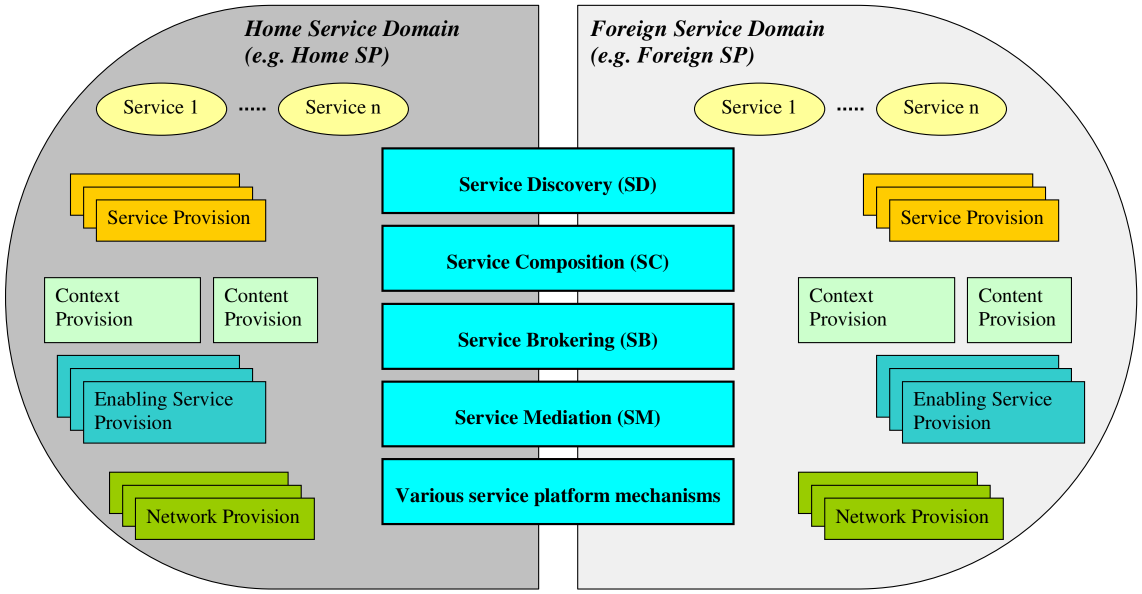

Figure 1. Service Platform Elements Used in Service and Information Roaming

This can be illustrated by an example service mechanism, namely service and information roaming (see Figure 1). Meaning of the service roaming is twofold: when a user enters a foreign service domain, he/she should have the access to the chosen set of home services and information, and the suitable services from the foreign domain (fitting / corresponding to the user profile, preferences, and service context information). At the same time the necessary information for the service execution (e.g., service context, **Identity management (IdM)**, **Quality of Service (QoS)** and **Authentication, Authorization, Audit, Accounting and Charging (A4C)** information) should be shared in a secure way. Just the necessary and previously cleared information is shared between the service functional blocks and service enablers.

Service roaming includes the collaboration of main service platform mechanisms, as **Service discovery (SD)**, **Service Composition (SC)**, **Service Brokering (SB)** and **Service Mediation (SM)**, context retrieval and management, etc. (see also Figure 1). Context information and knowledge is distributed, belonging to various platforms (service domains), and before sharing, it should be selected, secured, protected, and prepared for the delivery to the foreign service platform entities.

B. *Management of Context and User Profiles*

Enabling systems such as profile and context information management have to deal with both dynamic context information and with persistent profile data.

One of the key aspects in profile management is integration and operation with users' contextual preferences, since user's behavior and actions are situation dependent. By contextual preferences, we understand the users' preferences, which are linked to a particular situational context such as the user when driving a car or at work. The situational context of a user could for example be described with the user's current location, mood, the time of day, or any combination of different context variables.

Each application profile can be considered as a collection of sub-models that describe the application profile for different situations. An example application can have a sub-model representing a "user is at home" profile and a sub-model representing a "user is at work" profile. Thereto, firstly the context of a certain sub-model has to be added (e.g., through learning mechanisms or by the user with support of a profile editor). Secondly, the current context of a user has to be identified by the service platform. Thirdly, if the identified user context matches the context the sub-model depends on, the sub-model will be "activated" An application or service could then use these contextual user data and execute the resulting personalization actions.

In ubiquitous computing environments, we also have to deal with different applications and services. These all may use similar or even the same user data. From a user perspective, his/her user data should be shared by various applications to ensure a convenient single sign on. However, different applications usually use different semantics for similar or even the same user data. Thus reuse and sharing of user data between different applications and services is challenging. The use of Semantic Web technologies such as RDF(S) [10], OWL [9] and SWRL [11] and ontology management practices may help to overcome this challenge. Existing user model ontologies [4] are to be considered too.

Management of context and user profiles within the platform is performed on three logic levels: persistence, business and exposure [7], which interact through well-defined interfaces. Context information is stored by the **Context Broker (CB)**, which interacts with each **Context Provider (CP)** and exposes such data to the platform. Examples of context providers are presence servers, localization engines, environment sensors (see Subsection IIA).

The **Provisioning Framework (PF)** gathers, stores and manages the user profile (see Figure 2). As profiles are made up of very heterogeneous information, the PF will query other components such as the CB (for context data) and the device capabilities database. The **Profile Management Component (PMC)** can choose to store the gathered information in a database (static or semi-static data, such as usage statistics, or user history) or in memory (highly volatile data, such as the list of nearby devices).

The PMC has also the task to associate semantics with profile entries, to enable semantic access to the user profile. As a conclusion, a profile and context management component shall support management, inquiry and provision of (distributed) user data that depends on the user's situational context. Furthermore, the profile management component is to provide means for structuring and attaching semantics to the stored profile data. Last but not least, privacy and trust related issues are to be covered [2].

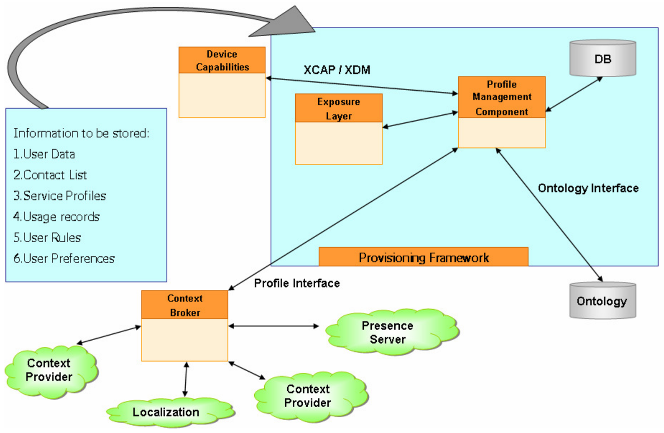

Figure 2: Provisioning Framework

C. *Context Modeling on Service Platforms*

Figure 3 shows the high-level context model. Based on the definition of context by Dey et al. ("Context is any information that can be used to characterize the situation of an entity") [3], we see an entity as something with a real existence, which is relevant for the service delivery and applications, e.g., a person, group, physical object, or computational component like a service. We add different attributes like *accuracy* and *probability of correctness* to express the quality characteristics of the context information. Examples of context are clickstream data, location information like cell IDs or GPS coordinates, remaining battery power, available networks, and individual or collaborative interests. Context can be either directly measured or derived from other data (e.g., aggregation of different cell IDs).

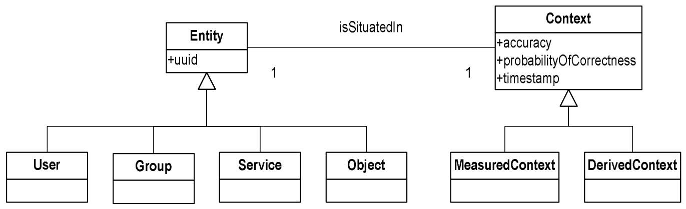

Figure 3: A High-level Context Model

An entity can be described with persistent user data such as name, address, phone number, etc. and with the current context of its situation. In parallel, user profile data can also be described with the persistent user data and specific context data in order to specify which situation the user data depends on. Usually, the user profile data includes basic information about the user, contact lists, information about groups the user is part of, subscription to services, user device information and user preferences. It may also include contextual annotations such as location, the time of day etc, describing the situation in which that persistent user data shall be applied.

IV. CONTEXT ENABLING AND USE

In this section, we illustrate importance of context use with a real life scenario, and show how context can be enabled and used in the mobile service platforms.

A. Usage Scenario

Scenario "Mobile Advertisements": When a user enters the cinema's foyer a selection of *individual* advertisements pops up on his mobile device. In order to choose which ads are most interesting to him/her the SPICE Provisioning Framework matches the cinema's offers with his/her individual interests. If he/she chooses to view at least one of the short advertisements, the cinema will grant him a 5% discount on the tickets. The discount can even be extended to 10%, if he/she chooses to buy one of the offered products.

The proposed above "Mobile Advertisements" scenario requires that the underlying service platform mechanisms function properly, i.e., when user enters the cinema's foyer, the following should happen:

- A4C functionality has to authenticate and authorise the user with the help of some identity management information. The service platform has to be prepared for auditing, accounting, and charging (completing at the end of the service life cycle).
- Service context information has to be retrieved and prepared for usage (collaboration of several service platform entities).
- SD and SC have to discover and to select service enablers and resources according to the service context parameters.
- Service brokering and mediation have to be used for selection and preparation of the best enablers for the service composition.
- SD and SC have to compose the service according to the user's service context definition
- The advertisement service has to be initiated and instantiated (or pooled for the later use).
- The service life cycle starts and the service execution environments have to use and process the information and knowledge, providing the wanted information and service to the end-user.
- According to the user's input and the service context information, the content ("personalized commercials") has to be provided to the user's device.
- The user then has to choose whether he/she wants to see the advertisement and receive an incentive or not.
- A4C and IdM enabling services have to collaborate in supporting the mobile commerce and completion of the service.

One could think of adding value functionalities in a scenario extension involving group management, assuming that the user interacts with his/her friends. In particular, the user would like to inform her buddies about his/her activities, so he/she uses a group instant messaging service to inform them. The user passes the information about the movie to her buddies, invites them to join, and provides the multimedia preview. Only the buddies whose context information allows interruption (employing an enabling service analyzing presence) are contacted/informed.

B. Enabling the Context

In a context aware system, typically many different context sources are available that can be classified by the type of context information that they produce. Single sources of the context information (e.g., pending emails) can be distinguished, as well as multiple sources of the same context information (e.g., location sensed by GPS, UMTS network, smart environments, etc.). Furthermore, these can (and in real world applications typically will) be distributed in different administrative domains (e.g., at home by your broadband provider, at work by your employer, and outside by your Telco provider). Dealing with heterogeneous contextual information coming from various sources and domains can be done by a Context Wrapper and Reasoner [8], supplying a context source component with a particular piece of context information through a well-defined ContextSourceInterface. Their goal is to provide a uniform interface to components or applications that use the context information, hiding the details of the underlying context-sensing mechanisms. Applications or users of this context can access this information by polling or by notification using publish/subscribe mechanisms. The description of the ContextSource is typically stored in a registry that can be searched by applications (discovery function of ContextBroker component). In principle, profile information delivered by the PMC (see Subsection IIIb) can be wrapped and also provided at a ContextSourceInterface.

The context management system facilitates context changes and reactions in a real-time fashion, and also provides means to store and log context data for post-processing and off-line analysis. The logged data allows for user behavior analysis, and knowledge inference and can feed learning and recommender components.

C. End Usage of Context on a Service Platform

The user can interact with the platform with a one-shot approach (“pull scenario”) or by subscribing to one or more service categories (“push scenario”), see Figure 4.

In a “pull scenario”, the **Advertising Module (AM)** gets the user context from the **Context Interpretation (CI)** module and asks the **Knowledge Inference module** for a ranked list of services. The list of services is sorted the usage history and the preferences set by the user.

In a “push scenario”, the AM does not start the query process immediately, but interacts with the **Context Monitor**. When the user context matches the trigger criteria, the AM receives a notification and starts the service discovery process following the same steps of the previous scenario.

The platform can access a semantic service catalogue, which contains meta-information about services and enables the execution of matching algorithms. At the same time, it can access the user profile, specifically the list of subscriptions to services and the context.

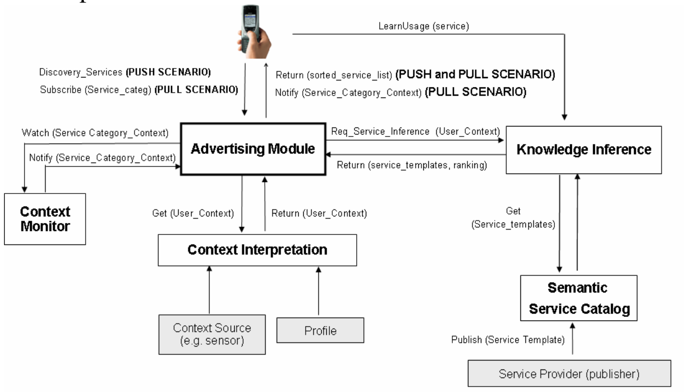
LearnUsage (service)
Discovery_Services (PUSH SCENARIO)
Subscribe (Service_categ) (PULL SCENARIO)
Return (sorted_service_list) (PUSH and PULL SCENARIO)
Notify (Service_Category_Context) (PULL SCENARIO)
Watch (Service Category_Context)
Notify (Service_Category_Context)
Advertising Module
Req_Service_Inference (User_Context)
Return (service_templates, ranking)
Knowledge Inference
Context Monitor
Get (User_Context)
Return (User_Context)
Get (Service_templates)
Context Interpretation
Semantic Service Catalog
Context Source (e.g. sensor)
Profile
Publish (Service Template)
Service Provider (publisher)

Figure 4. Attentive Services Architecture

When the user context matches some criteria (e.g., the user approaches an airport), his device shows a list of available services (e.g., mobile check-in, flight info, hotel booking, car rental). The order in which the services are shown is determined by the usage history and by preferences explicitly set by the user.

The user can benefit from the offered services, because they are highly-customized and require very little interaction: for example, if the user agenda contains an entry regarding a business meeting in Paris, the hotel booking service will offer as first choices the hotels nearby the meeting venue.

V. CONCLUSION

We presented foundational ideas on how to acquire, represent, employ and use context information in mobile service platforms. Specifically, the outlined principles and challenges form a context awareness basis for the mobile service platform developed in the IST project SPICE.

ACKNOWLEDGMENT

A. V. Zhdanova thanks Mirko Presser (University of Surrey) for pointers to useful literature. M. Marengo thanks Massimo Valla (Telecom Italia) for useful discussions.

REFERENCES

[1] M. Beigl, A. Krohn, T. Zimmer, and C. Decker, “Typical Sensors needed in Ubiquitous and Pervasive Computing”, *First International Workshop on Networked Sensing Systems (INSS)* 2004, *Society of Instrument and Control Engineers (SICE)*, vol. 04 PR0001, ISBN 4-907764-21-9, 2004, pp. 153-158.

[2] O. Coutand, M. Sutterer, S. Lau, O. Droegehorn, and K. David, “User Profile Management for Personalizing Services in Pervasive Computing,” *6th International Workshop on Applications and Services in Wireless Networks*, Berlin, Germany 2006, to be published.

[3] A. K. Dey, D. Salber, and G. D. Abowd, “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications”**,** *Human-Computer Interaction (HCI) Journal, Volume 16 (2-4)*, 2001, pp. 97-166.

[4] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. v. Wilamowitz-Moellendorff, “GUMO – The General User Model Ontology,” *Proceedings of the 10th International Conference on User Modeling (UM'2005)*, Edinburgh, UK, 2005, Springer-Verlag Berlin Heidelberg, LNAI 3538, 2005, pp. 428-432.

[5] P. Korpipää, J. Häkkilä, J. Kela, S. Ronkainen, and I. Känsälä, “Utilising context ontology in mobile device application personalization,” ACM International Conference Proceeding Series; Vol. 83 *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia,* ACM Press, 2004, pp. 133-140.

[6] D. Pfoser, E. Pitoura, and N. Tryfona, “Metadata Modeling in a Global Computing Environment,” *GIS’02*, November 8-9, 2002, McLean, Virginia, USA., ACM Press, 2002.

[7] Three tier architecture [Online], 2006. Available: http://en.wikipedia.org/wiki/Three-tier_(computing)

[8] H. van Kranenburg and H. Eertink, “Processing Heterogeneous Context Information”, *Proceedings of 2005 Symposium on Applications and the Internet, Next Generation IP-based Service Platforms for Future Mobile Systems workshop, (SAINT 2005)*, ISBN 0-7695-2263-7, 2005, pp. 140-143.

[9] W3C: Web Ontology Language (OWL) [Online], 2004, Available: http://www.w3.org/2004/OWL/

[10] W3C: Resource Description Framework (RDF) [Online], 2004, Available: http://www.w3.org/RDF/

[11] W3C: A Semantic Web Rule Language (SWRL) Combining OWL and RuleML [Online], 2004, Available: http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/