Service Frameworks for Mobile Context-aware Applications

Johan KOOLWAAIJ[1], Patrick STRATING [2]
Telematica Instituut, Drienerlolaan 5, Enschede, The Netherlands
Tel: +31 53 485 0 485, Fax: +31 53 485 0 400
Email: [1]Johan.Koolwaaij@telin.nl, [2]Patrick.Strating@telin.nl

Abstract: Providing an easy-to-use service framework for mobile context-aware applications is a real challenge. This paper describes how context information in mobile 3G networks can be unlocked via web services. We describe a service framework that builds on top of these web services, to enable the rapid development of applications that are aware of the context of the end-user, are able to reason with context information and to serve the end-user with an offering of functionality and services that are relevant for him or her as a unique individual in a specific context, e.g. role, location or activity.

1 Introduction

Many people can no longer imagine daily life without a mobile communication. GSM technology – now so familiar that we often refer to the mobile phone as 'the GSM' – is now making way for next generation (3G) technologies like GPRS and UMTS. These networks, which, like the Internet, are package-switched, enable new voice and data services. In addition, the networks are becoming more intelligent as they have access to (or grant other parties access to) information about the user, such as the whereabouts of the user, or his personal or activity profile. With the aid of that information, applications can be developed that adapt to the user, his preferences and the (changing) environment, in other words, that are context-aware.

A context-aware application could, as an example, provide a business traveler with a route covering all his appointments for the day, with a stop in an interesting museum during the lunch break, based on his personal profile. Once at the museum, he would automatically receive further information about the museum, its collection and the services the museum offers – once again tailored to his requirements and interests. Similar applications can be envisaged for new employees of large organizations who want to learn to find their way around the company, for service engineers who need information en route about the objects to be repaired, or for emergency departments, including police and ambulance. Other examples can be taken from business to employee services. Employees increasingly function as flexible workers capable of performing different tasks relevant to their company. Allowing flexible switching between various tasks creates new ways of human centered work scheduling. In order to support the employee with the appropriate information and services, supporting services must be readily tuned and adapted to the tasks, context and devices of the employee. This will not only render more efficient work processes for the company, but also broaden and enrich the work of the employee, and increase his value as a knowledge worker.

The WASP project (Web Architectures for Services Platforms) proposes a web services based framework to facilitate and speed-up the development and deployment of context-aware integrated mobile speech and data applications [1]. A web service is a software application, whose interfaces are defined, described and discovered using XML based standards and that supports direct interactions with other software applications using XML

based messages via internet-based protocols. Such web services can be used to unlock context information from different sources and to make this information available in an easy and standardized way. The concept of web services is gaining momentum in the sense that both the understanding and the popularity of this concept are increasing. However, as it is an emerging technology, we can still identify a number of open issues, especially issues that emerge in context-aware applications, including the semantic richness of the web service descriptions and event handling in a request-response model. In the WASP project, one of the goals is to validate the claim that cross-platform application development, in particular regarding flexible mobile context-aware applications, is easier and faster when web services technologies (SOAP, WSDL, UDDI, XML) are used.

1.1 Scenario: Grand Tourism in the year 2003

The service framework that is developed within the WASP project is generic by nature. That is to say that by using the framework context-aware capabilities can be added to new or existing applications in different domains, from business-to-employee services to wireless services and services in tourism. However, to visualize the capabilities of the service framework we have described a scenario for context-aware applications that assist tourists in navigation in unfamiliar environments, in suggesting interesting places to visit, and in communication with people, places and services in their vicinity. This scenario guides the demonstration applications that we are currently developing within the WASP project. In the scenario an American family with two children, Liddy and Luke, visits the Netherlands for holidays.

Dynamic navigation: On a Thursday afternoon, the Stephenson family arrives in Enschede by their own transport: a small size MPV rented from Schiphol Airport. The rental car was top ranked in the suggestion list of the automatic car rental and reservation system, and really suits the needs and wishes of the Stephenson family. They have booked a hotel in the city center of Enschede. Since Mr. Stephenson has never been in the eastern part of The Netherlands before, he uses an on-line navigation guide to get to his destination. The suggested route matches the usual driving style of Mr. Stephenson during holidays: use the main route, but suggest touristic detours. The guide correctly informs him about construction works in the south of the city center and proposes an alternate route without obstacles.

Meeting friends: During the drive to Enschede, son Luke already found out that the daughter of one of his father's old friends, Mr. Vaneden, is also in the Netherlands, because she is on his 'buddy list' and he received a notification once they were in each other's vicinity. He calls her by simply selecting the call option that comes with the notification, and she tells him that her familiy is staying in a hotel in the picturesque town of Ootmarsum, about 20 km north of Enschede. He proposes to his family to dine with them. Since they all like the idea, Luke suggests a date in the schedule of the Vaneden family. The place to dine will be decided by Luke later, but he promises that it will be a restaurant that meets the expectations and tastes of all of them.

Interesting Places: The hotel rooms are convenient and at the quiet side of the building. Although it is late in the afternoon, Liddy wants to have a 'sneak preview' of the touristic attractions of Enschede. It was her father's idea to visit Enschede because of the Aviation Museum and some local railroads with running steam engines, but she hopes that there are also some other attractions as well. To search for interesting information, Liddy uses a terminal provided by the hotel, because this terminal has better interaction capabilities than her PDA. Her portal offers a personalized guide to the touristic attractions of Enschede. To tailor the guide to the visitor's interests, the site asks the visitor to explicitly describe his interests or to give permission to obtain (parts of) the profile of the visitor. Liddy chooses

for the last option. Given the information from the profile, the portal supplies Liddy with an interactive map with all relevant points of interests in Enschede and the surrounding area. Liddy chooses for all museums within walking distance. A map is shown with a route, starting from Liddy's current position, along three museums: the Rijksmuseum Twenthe (Art), the Natuurmuseum (Nature) and the Jannink museum (Textile). The maps shows estimations of the walking time and visiting time, based on her walking speed and the museum tour lengths of the different museums. Liddy is also notified that there will be an art market on Saturday at the Van Heek Plein starting at 10:00.

2. Objectives

The creation of usable and feasible mobile applications will largely depend on the availability of flexible service frameworks. In particular when these mobile applications need to be aware of the context of the mobile end-user. The user's context plays an important role in determining what is relevant for a mobile application to offer.

One of the early definitions of this concept (Schilit and Theimer, 1994) states that context-awareness is the ability to adapt according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time" [5]. In our view, context-awareness has many more dimensions than only location and time. A typical user context can be divided into several context elements: environmental elements, personal elements, task-related elements, social elements, historical elements, and spatio-temporal elements [9]. To serve a user in his context with relevant service offerings is a complex task when all these context elements have to be taken into account. As an example, we can imagine a business traveler in Amsterdam (spatial), who is heading for his hotel (task), likes public transport more than taking a taxi (personal) and travels with his wife and one-year old child (social) while it is a rainy day (environment). These context elements can rarely be obtained from one single source. More naturally, these elements can be obtained from a wide spectrum of sources, distributed over different parties and locations, but in a generic way and accessible via well-described interfaces that are exposed to a public network, such as Internet. A service framework is required to deliver value-added services on top of existing low-granular services and knowledge sources [4].

Such a service framework has to support the whole service lifecycle including definition, implementation and deployment in order to speed up the service creation process. Application developers will be able to choose from modular services when creating new mobile applications. Some of those services might offer functionality (e.g. call control, geographic intelligence), whereas others offer (context) information and knowledge. The service framework will make it easier for providers of services to offer and manage their services, while for the users of the services searching, finding and purchasing these services becomes easier. In addition, through the use of the web services paradigm, these services will be able to utilize the intelligence of the mobile network, something that is not easy at present. This paper will address the challenges in providing a service framework for mobile, context-aware applications.

The service framework needs to be transparent, dynamic, platform-independent and highly tailored toward convenience for developers. The objective of the WASP project is to develop such a service framework for mobile, context-aware services. This framework consists, on the one hand, of a mobile 3G platform that unlocks the information and capacities in the network by means of web services. On the other hand, the framework consists of an application platform that dynamically connects the user with information and

services that are relevant to him. The WASP service framework has the following characteristics:

- Applications are independent of the underlying platform or the network used, and can operate in multi-operator and multi-vendor environments
- Seamless integration between networks and services is possible based on web services with clear, standardized interfaces for information exchange
- Services, and multimedia services in particular, can be adapted to the capacities of the network and the end user's mobile device
- It offers a personalized service environment [10]
- Intelligent searching for relevant services in a broad and dynamic range of services
- Combination of voice and data services

In the past, standards organizations such as OSA and Parlay have called for such benefits. However, in practice, integration with interfaces according to these standards remains a complex issue, that requires years of study and experience. In addition, the relevant knowledge is hard to transfer. This hampers rapid application development. Because of their transparent approach, web services can potentially access a much larger market. And because of the worldwide standardization of the publication, description and invocation of services, knowledge regarding these services is transferred more easily. The WASP project investigates the possibilities of this new approach in the context of mobile, context-aware services.

3. Implementation

Within the WASP project we have developed a service framework that is basically a collection of generic building blocks for context-aware applications. It offers generic functionality in the area of context-aware logic and profile management to applications that run on top of it, herewith easing the development of context-aware integrated mobile speech and data applications. All building blocks are accessible via web services with standardized interfaces.

A typical interaction pattern of an application using the framework is visualized in Figure 1. The service framework directly integrates with mobile 3G networks to obtain context parameters and to access network capabilities, such as call control and charging. These network capabilities are also accessible via Parlay-X web services, standardized by the Parlay community [2].

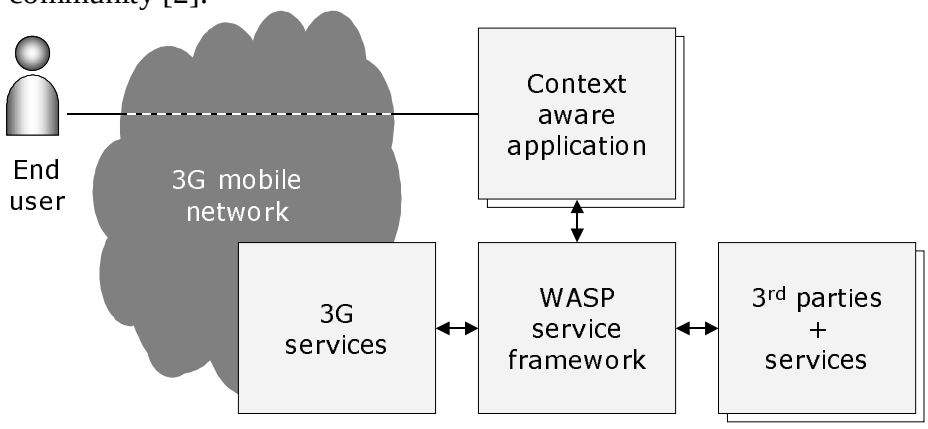

Figure 1: The interaction pattern of a typical context-aware application with the WASP service framework

The service framework also integrates with 3rd parties and their services. Since one of the functionalities of the framework is to connect people to places, that is to let them communicate with locations (buildings, services, other people) in the proximity, the framework also has a registry that contains potentially interesting locations, the meta-data describing those locations, and the interaction patterns for the services offered by those locations. The registry then helps to find relevant locations and to integrate with the services they offer. But a 3rd party service can also offer generic functionality such as calculation, searching, mapping or navigation, e.g. the Microsoft MapPoint service offering programmable maps and other location intelligence that can easily be integrated in other applications [3].

As an example, an end-user may use a specific application to find the most nearby ATM that accepts his card. Without having to specify his current location or the name of his home bank explicitly, a map is presented by the application that shows the route from his current location to the closest ATM. Under the hood and invisible for the end-user, his location is obtained from the 3G context service, his home bank has been retrieved from his personal profile, the registry has been searched for ATM accepting this specific card, and the MapPoint service is used to generate the interactive map that guides the user to his goal, extended with push-to-call functionality to call his bank in case something goes wrong.

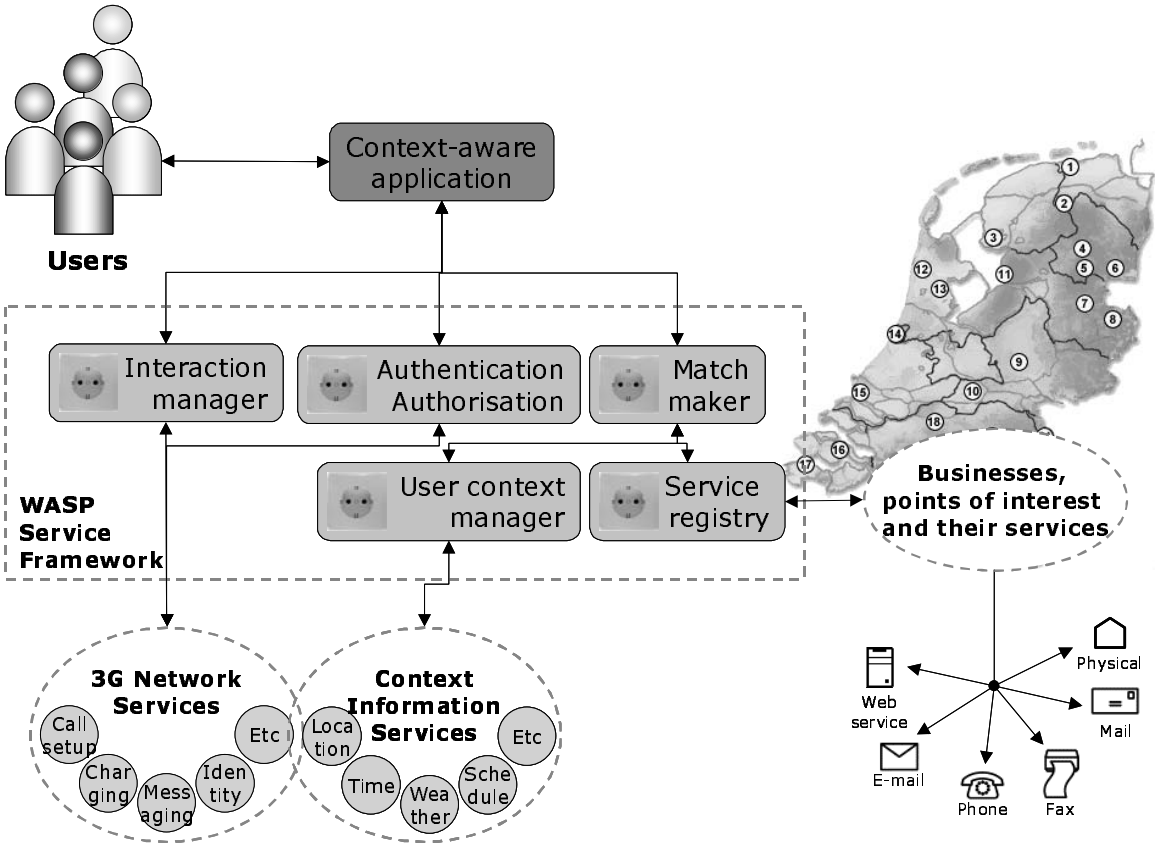

Figure 2: An architecture impression of the WASP service framework

As described earlier, the service framework is a collection of building blocks for context-aware applications. Each building block is exposed as a web service to be used by context-aware applications or other building blocks. Figure 2 depicts an impression of the key building blocks in the framework and the interaction between those blocks. The building blocks allow context-aware applications to easily identify and authenticate the user, to obtain context information about the user, to find businesses, points of interest and/or services that are of interest to this particular user, and to interact with the offered services via e-mail, telephone or web services. From a service provider perspective, the framework

allows to register and manage information about their businesses, points of interest and the services they offer.

Basically, the framework performs three tasks to make the applications built on top of the framework context-aware. 1) It can obtain and maintain context information for a mobile user. 2) It assists in reasoning with context, and selecting relevant services and information given the user's context. 3) It enables easy interaction with relevant businesses, services or other users.

The context information is obtained from heterogeneous context information services by the user context manager. The context manager can retrieve information about user status and location from the GPRS or UMTS networks via Parlay-X web services, it can retrieve information about the user's environment from external sources such as weather or traffic information services, or from personal information sources such as his/her schedule. The user context manager is keeping track of the changing context of a user during a session. It can provide alerts based on the user's schedule or location. Furthermore, it serves as the context knowledge source for the matchmaker component that supplies the user with a relevant service offering. This service offering is selected from the service registry that holds (links to) information on businesses and their services. The matchmaker has the intelligence to bring down the huge list of available business and services to a shortlist of the most relevant ones for a specific user in his/her context. When a user wants to interact with a specific service or with another mobile user, the interaction manager assists in finding the best way for interaction and assists in interaction. To give an example, a user logs in to a business web site using a desktop PC. This web site is context-aware (and thus connected with the framework depicted in Figure 2) and offers a service to call their sales department with one single button click. Although the user is currently operating a desktop PC, the interaction manager sets up a connection between his mobile phone and the sales department, without the user having to punch a number on his mobile phone.

4. Application

Systems for obtaining information about the surrounding environment are the most popular type of context-aware applications. This was experimented with already in the ParcTab system and in other projects in the beginning of the 90's. The Chameleon project at the University of Toronto investigated how palmtop computers could become aware of their location and orientation and give information about physical objects in their environment.
Guides of different kind using location information are actually the commercially best-covered area of all context-aware applications. Mobile phone operators have been prototyping for some years now and offer location based information services. For example, O2 offers a service under the name 'Spotter' that allows people to request the location of friends and relatives from the mobile network via SMS [7]. Location is usually determined using a network of satellites (GPS) or a network of antennas, either wide area (GSM/GPRS) or local area (e.g. Bluetooth) [11,12]. Research in the WASP project is focused on outdoor localization techniques, based on GPS or the GPRS network.

A demonstration application that uses the functionality from WASP service framework will be developed together with regional tourism offices in the Netherlands. This application will support the scenario described in section 1.1 and, more specifically, it will offer the following context-aware capabilities:

1. Points of my interest: information services around buildings and locations that are of interest for tourists with specific tastes and interests, with for example historical information about monumental buildings, access to digital image libraries with old postcards, visitor information, but also interactive services such as museum catalog browsing or table reservation in restaurants.
2. Navigation: dynamic and interactive maps and city navigation, putting relevant locations and people on the map, and offering easy access to services and communication with people and places. (For an example see the figure with the screen shot of a mobile phone).
3. Notification: services that notify the end-user of events that are close (measured in time, place or degree of interest).

These capabilities can be delivered relatively easily using the building blocks in the WASP service framework. All building blocks are exposed as web services, which makes them self-describing – to a certain extent. To improve the semantic richness of these descriptions we use semantic web technology, including DAML-S [8] to relate the descriptions (inputs, outputs and meta-data) of the services to higher order ontologies. The framework enables service providers to describe their services in detail (capabilities, functionality, constraints, context, etc.), and to bring those service descriptions in correspondence with existing ontologies. On the other hand, it enables the searching parties to perform a subtle search, by using constraints, relations between concepts, approximate matches and semantically rich queries [6], which delivers a more manageable result set

5. The road ahead

Although the concept of context-aware applications has been around for more than a decade, the first serious, commercial applications are brought out just now. All these applications share the functionality to retrieve context information from heterogeneous sources, to reason with this context information, and to offer services and information that are relevant in this context. They also share the same challenges: how to respect the privacy of the end-user, how to secure the transmission of context data, how to solve event handling needed for notification, and how to retain a scalable solution for context monitoring for large volumes of users. There is obviously a need for a reusable solution that responds to those capabilities and challenges, that is easy to use for a broad array of applications.

For this reason, we have presented the service framework developed in the WASP project that supports the development of context-aware applications by delivering the generic functionality that forms the basis for all context-aware applications. The building blocks in the framework provide a reusable solution for context information retrieval, reasoning with context, and performing actions driven by context, together with solutions for authentication, security and scalability. Application developers can enhance new or existing applications with context-aware capabilities by simply using the generic, domain independent building blocks in the framework. These building blocks have a plug-and-play nature by using the web service paradigm for integration purposes. The framework allows

for dynamic composition of services and reasoning with context elements by combining ideas coming from the web service paradigm with those from the semantic web vision.
The validity of the framework is being assessed in the scope of tourism applications. New context-aware applications are being deployed and existing applications can be retrofitted with context-aware capabilities that open up the way to easy to use, personalized functionality for those applications.

More information on context-awareness in general or on the service framework in particular can be obtained from the WASP document archive [1].

6. References

[1] WASP, Web Architectures for Services Platforms, http://wasp.freeband.nl

[2] Parlay-X, Web Services in the Parlay Environment, http://www.parlay.org

[3] MapPoint, Microsoft MapPoint Web Services, http://www.mappoint.net

[4] A. Harter et al.,The Anatomy of a Context-Aware Application, Mobile Computing and Networking, 2002, http://www.uk.research.att.com/pub/docs/att/tr.1999.7.pdf

[5] B.N. Schilit and M.M. Theimer, Disseminating active map information to mobile hosts, IEEE Network 8(5): pp. 22-32, 1994, ftp://ftp.parc.xerox.com/pub/schilit/AMS.ps.Z

[6] Extending S. Pokraev et al., UDDI with context-aware features based on semantic service descriptions, The 2003 International Conference on Web Services (ICWS 2003), Las Vegas, USA.

[7] O2 spotter, a service to locate friends and relatives, http://www.o2.nl/spotter

[8] DAML-S, a DAML-based Web Service Ontology, http://www.daml.org/services

[9] S. Pokraev et al., Context-aware services, WASP state of the art report, Telematica Instituut, 2003, in progress.

[10] G.C. van den Eijkel, M.M. Lankhorst, H.van Kranenburg, M.E. Bijlsma, Personalization of next-generation mobile services, Position paper Wireless World Research Forum, Munich, Germany, 2001.

[11] A. Peddemors, M.. Lankhorst, & J. de Heer, Presence, location and instant messaging in a context-aware application framework 4th International Conference on Mobile Data Management (MDM2003), Melbourne, Australia, 2003.

[12] N. Boertien, Location based services: services and technologies, Virtuele Haven report series T3.D9, Telematica Instituut, 2002. https://doc.telin.nl/dscgi/ds.py/Get/File-19920