Context-Awareness in MobiLife

R.Kernchen, David Bonnefoy, A. Battestini, B. Mrohs, M. Wagner, and M. Klemettinen

***Abstract*— The IST project MobiLife (IST-2004-511607) is approaching Context-Awareness from the perspective of mobile applications and their enabling components. It envisions a highly distributed and loosely coupled solution in order to exchange context information on different semantic context levels between applications, enabling components and raw context sources. Semantic meaning of the context information exchanged is added via distributed ontology's attached to it. The definition of the mechanism and interfaces is described by the MobiLife Context Management Framework. Actual context data used within MobiLife applications is provided and defined by a number of specific context providers. New context providers can be easily introduced based on the generic definition of them. The paper aims to provide the overview to the MobiLife context-awareness approach and describes its usage by the means of two MobiLife applications.**

I. Introduction

The goal of the MobiLife project, one of the Integrated Projects in IST-FP6, is to bring advances in mobile applications and services within the reach of users in their everyday life by innovating and deploying new applications and services based on the evolving capabilities of the 3G systems and beyond. The vision of the project is that mobile and wireless applications and services become everyday things in a world where people shift continuously between different (blurring) roles, in participating different groups, using multiple devices, with different modalities, over different networks. The consequence is that mobile and wireless applications are increasingly context-aware, that is, informed about the context of the end-user and able to respond properly to changes in the user context.

To realise such a vision, the MobiLife project has to overcome a number of challenges that are common to the generic class of context-aware mobile applications, including (a) the handling of context information, from context modelling to context distribution and gathering and smart context inference approaches, (b) the provisioning of services that are relevant to an end-user in a given context. To enable the rapid development of context-aware applications, the re-use of generic components and the standardisation of syntax and semantics to describe context information, strong architectural support are necessary that is capable of alleviating the hurdles to the development of context-aware applications.

This paper is organised as follows. Section II introduces the different dimensions of context in MobiLife and Section III describes the MobiLife Context Management Framework (CMF). Section IV gives a more in-depth view on how context information is been published in the CMF through the Context Provider concept and examples of actual realised Context Provider. Section V gives the description of the Multimedia Infotainer and the Context Watcher application related to the usage of the Context Provider. Section VI draws the conclusions.

II. Context-Awareness in MobiLife

Where possible, we follow the international literature on these topics, e.g., the widely accepted definitions by Dey and Abowd for the context: "Any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." [1].

MobiLife considers Context-Awareness from three perspectives in order to address particular challenges in providing and using context for individuals, groups and more general on how to model context from a broader perspective. The general approach is necessary in order to provide a common framework on how to exchange, describe and manage context information. Therefore throughout the whole MobiLife project wherever context is used, it is supported by the concept of the Context Management Framework (CMF) as described in more detail in Section III.

Nevertheless different challenges were addressed from the perspective of an individual and groups. Context-Awareness for individuals is focusing on the adaptation aspects of services for individuals. This includes research on user interface adaptation as described in [2] and on contextual personalisation [3]. The objective here is to show how context can be used to personalise the user's service and how service content is best presented to the user based on the location, user activity and available devices in the user's environment. Furthermore, the system can recommend the usage of certain modalities for content

.

This work has been performed in the framework of the IST project IST-2004-511607 MobiLife, which is partly funded by the European Union.

Ralf Kernchen is with University of Surrey, UK (e-mail: R.Kernchen@surrey.ac.uk).
David Bonnefoy is with Motorola, France (e-mail: David.Bonnefoy@motorola.com).
Agathe Battestini and Mika Klemettinen are with Nokia Research Center, Finland (e-mail: {agathe.battestini,mika.klemettinen}@nokia. com).
Bernd Mrohs is with Fraunhofer FOKUS, Germany (e-mail: bernd.mrohs@fokus.fraunhofer.de).
Matthias Wagner is with DoCoMo Eurolabs, Germany (e-mail: wagner@docomolab-euro.com).

delivery and also generate application based recommendations. For example, a wellness-aware fitness service can use the system to generate training proposals based on the user's training history. Further applications can profit from direct learning from user behaviour. The user interface adaptation therefore offers the collection of active user input (where the user interacts actively with a service through the user interface) and passive user input (collected sensor information such as heart rate, location and further physical parameters).

Group Awareness in MobiLife refers to the use of context information related to a group that enables the provisioning of ubiquitous applications and services in order to address the groups concerns and needs [4]. The focus on groups lies on mobility and therefore dynamic behaviour of groups. Usually groups are formed when certain interests of individuals fit together, therefore a group of interest or maybe a group with a common goal. In order to enable a system to support groups, the system has to provide a group management functionality what then allows to represent the group within the system. The group management therefore is able to logically create and dispose groups, to add or remove group members in order to provide the group with information and perform certain actions for the group based on the current group context and the group's preferences.

As mentioned before all these functionalities are based on the common MobiLife CMF in order to retrieve and make use of context information on different context levels. The next section will describe the CMF in general and Section IV will describe which particular context information is technically available at this point of time.

III. MobiLife Context Management Framework

This section explains the MobiLife Context Management Framework (CMF), which is the MobiLife approach for the discovery of, the exchange of, and the reasoning on context information, in such a way that context information can easily flow from one provider to multiple consumers, and from multiple providers to one consumer, in order to build smart constellations of context providers that finally can produce high-level situational information. This high level information is then built upon tiny bits of context information from heterogeneous sources, where heterogeneity has different dimensions, from syntax and semantics to transport, security and protocols.

The main tasks for the MobiLife Context Management Framework (CMF) are:

- Enable the discovery of context providers
- Standardise context exchange between providers and consumers
- Share common understanding about context information elements
- Support plug-and-play reasoning with context information
- Support the construction of different constellations of context providers to provide high-level situational information, e.g. the LEGO® approach, or more technically, pipe-and-filter.

Important characteristics of context that have been kept in mind while developing the CMF are:

- Context information typically comes from diverse and heterogeneous sources.
- Storing all context information is impossible and would be almost the same as caching all the data that is routed over a server in the Internet.
- Scalability, data management and privacy issues are important and should be taken into account from day one.
- The distributed nature of the components in the CMF, e.g., context providers might reside on the mobile phone, on the mobile phone of someone else, somewhere in the network, virtually everywhere.
- Context information is dynamic and is bound to change frequently and may have a limited validity.
- Context information may be incomplete, inconsistent, and/or erroneous.
- The relevance of contextual information depends on the application and situation at hand.

We define the MobiLife CMF as a set of components, which are connected at build time, that together provide for the relevant context information, both by sensing and interpreting mechanisms. The interpretation may internally implement several different reasoning mechanisms, but even though the inner workings of such instances may differ, they all provide the same interfaces and also require the same basic set of interfaces.

The key components or building blocks of the CMF are depicted in Figure 1 and described below.

Context Consumer (CC): A software entity that uses the context provider interface, as a communications endpoint, for obtaining contextual data.

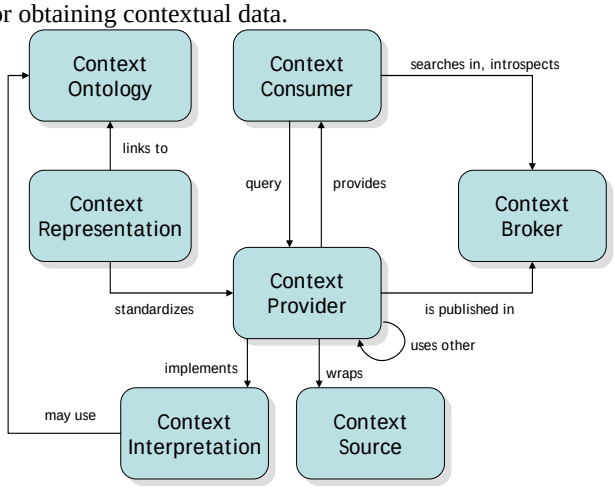

Figure 1: Context Management Framwork (CMF)

Context Provider (CP): A software entity that produces new context information from internal or external (context)

information. A context provider has exposed interfaces to provide context information to context consumers. These interfaces adhere to standards defined in the CRF. It is registered in the CB so that context consumers can discover and introspect it. The internal working of a context provider is usually hidden, but may include context aggregation, caching, prediction and other processes. Note that the term context provider is also used as a role, so that other components like a reasoner or storage component can take this role.

Context Source (CS): An information store that delivers potential context information. A context source can be wrapped as a context provider that has the correct interfaces to deliver this information to context consumers. The context source is usually not under MobiLife control, it is a given, and does not adhere to the CRF. Examples of a data source are a GPS receiver or a rain sensor. The simplest form of a CP would simply wrap these CS, and provide their information in a way that adheres to the CRF, a more complex CP would add some logic to derive higher-order information e.g. country instead of raw GPS coordinates.

Context Broker (CB): A registration and lookup service to enable the discovery of various context providers, their interfaces, the context information elements they can supply and the entities that play a role in (or: are related to) these context information elements.

Context Representation Framework (CRF): A set of specifications describing the context provider advertisements, the context provider interfaces, a format to exchange an atomic context information element and collections thereof, as well as the interfaces to exchange context information.

Context Ontology (CO): A service that provides access to the MobiLife ontology. The ontology itself can however be modular and/or distributed in nature. The ontology describes the qualitative context information elements and their interrelations, and serves as a semantic common ground between different context providers, as well as a possible knowledge base for reasoning.

Context Interpretation (CI): The collection of components that process contextual information in order to provide semantically flavoured descriptions of contextual information. This may include feature extraction, selection, classification and other aspects.

IV. The MobiLife Context Provider

This section gives a description about the concept of the Context Provider as defined and used within MobiLife. As has been mentioned before the Context Provider has the role to provide context through a defined interface towards Context Consumers. It can gather context information directly from context sources for example sensor information or information sources like the internet. Context Provider can be Context Consumers at the same time to aggregate context information on different levels and to provide context in a granularity as a certain service is requesting and using it. In the next two subsections an overview about a possible internal structure of Context Providers is given and additional a description of a number of currently available Context Provider and their context information is provided.

A. *Context Provider description*

Due to the fact that Context Providers can be versatile and exchangeable the internal structure and the design of Context Providers is not fixed as well. Still a general structure can be given to frame which tasks Context Provider do fulfil. Figure 2 shows such an internal design of the MobiLife Context Provider. As can be seen at the bottom of the illustration a Context Provider takes input from direct context sources (e.g. sensors, internet information) and other Context Providers.

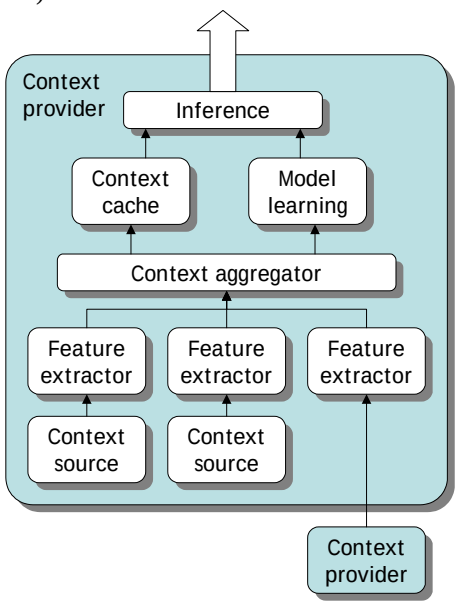

Figure 2: MobiLife Context Provider

In general the first step is the extraction of relevant context information for further processing within a Context Provider. Specific extracted features are aggregated in a storable and usable representation inside the Context Provider. Usually a Context Provider comes with the functionality to cache or store context information for processing. In the case Context Providers are design to learn based on context information certain learning models will be part of the Context Provider. The final process in order to generate new context information is the inference mechanism (e.g. rule based reasoning). The inferred new context information is been advertised throughout the system for usage in services or further processing.

B. *Available Context Providers*

From the theoretical definition the MobiLife project was deriving a number of implemented context providers as basic for MobiLife context-aware applications. Although

these several context provider realisations are by no means complete, this section actually gives a very good impression how useful and flexible the approach is. A brief overview about available Context Providers is given together with a rough overview about the context data they provide. Detailed programmers information is available from the Web service description [6]

1) *Location CP*

The Location CP provides users' location information. The location information could be in terms of cell id, GPS coordinates or readable street address. The Location CP gets location information as cell id and GPS coordinates. Using this info it then tries to identify the actual street address. All these location information including the original positioning information is made available to all authorised parties.

Parameter structure:

- *cell*: the cell info.
- *waypoint*: the GPS coordinates
- *address*: the postal address
- *cluster*: the location cluster the user is in
- *trajectory*: the starting and the ending path of the user.
- *all*: all location specific context

2) *Environment CP*

The Weather CP delivers environmental related information of the location specified by the name of the city and the country. The contextual data provided comprises a description of the current weather situation as well as a forecast for the next few days, covering all major regions such as Africa, America, Asia, and Europe. Besides actual weather information such as temperature (measured and RealFeel), wind (speed and direction), precipitation (snow and rain), UV index, humidity, pressure, visibility and a textual description of the current conditions, one can also retrieve prognosis of those elements (e.g., the expected maximum temperature for tomorrow).

Parameter structure:

- *Weather*: detailed weather information of the specified location
- *sun*
- *moon*
- *timezone*

3) *Preference CP*

This CP delivers individual or group-related personalisation information or service specific preferences. Such information can be used by any MobiLife services to personalise services based on users' preferences.

Parameter structure:

- *record*: Personal details of the user
- *global*: global preferences for all services
- *services*: preferences related to specified service.

4) *Wellness CP*

The Wellness CP provides various health related information. Targeted specially for sport fanatics, this CP provides a history of heart rate, speed and distance information during cycling, hiking or skating.

Parameter structure:

- *hearthrate*: the heart rate of the user
- *walkspeed*: the walking speed
- *walkdistance*: the distance the user walked.

5) *Presence CP*

This Context Provider provides context information on the presence - namely 'userAvailability' (busy, away, online, etc.) and the 'userMood' (Happy, Sad) of users. In addition to these two presence categories, this CP also provides custom presence information specified by the user.

Parameter structure:

- *userAvailability*: the availability of the user
- *userMood*: the mood of the user
- *customPresence*: the custom presence specified by users
- *presence*: all above presence information of the user together

V. MobiLife Context Usage Examples

One major focus of MobiLife is to bring advances to individuals and groups in their everyday life through the design and development of new application and services. A number of interesting applications were identified which are using a set of context information as envisioned in the prior described sections. This section will explain briefly two of the several applications appealing to be most interesting from the background of usage of context information respectively being context source and providing context information. The Multimedia Infotainer will serve as example for possible context sources and the Context Watcher example describes the dimension on how context can be perceived using the MobiLife Context Management Framework and their Context Provider.

A. *Multimedia Infotainer*

One of the MobiLife goals is to make it easier for users to make the best use of the complex environment that surrounds them due to advances in technologies. One aspect of this is the management of the multiple devices that are made available to them to display and interact with multimedia content: now their mobile devices, audio car kits, etc. and soon connected home theatres for instance. MobiLife has developed a context-aware multimodality framework that allows selecting the best matching device and modality for output and input in the current user context. This framework has been demonstrated through the Multimedia Infotainer application, which allows users to receive personalised content on multiple devices and

through multiple modalities. For instance, when the user is in his car, incoming messages would be spoken to the user instead of being displayed as text for safety reasons.
The multimodal framework is itself a Context Provider: it discovers devices in the vicinity of the user, and makes this information available as context. This context information, aggregated with other context information, can be used to infer higher-level information such as the location or activity of the user.

B. *Context Watcher*

The Context Watcher is a mobile application developed in Python, and running on Nokia Series 60 phones. Its aim is to make it easy for an end-user to automatically record, store, and use context information, e.g. for personalisation purposes, as input parameter to information services, or to share with family, friends, colleagues or other relations, or just to log them for future use or to perform statistics on your own life.

The Context Watcher application is able to record information about the user's

- Location (based GPS and/or GSM cell based)

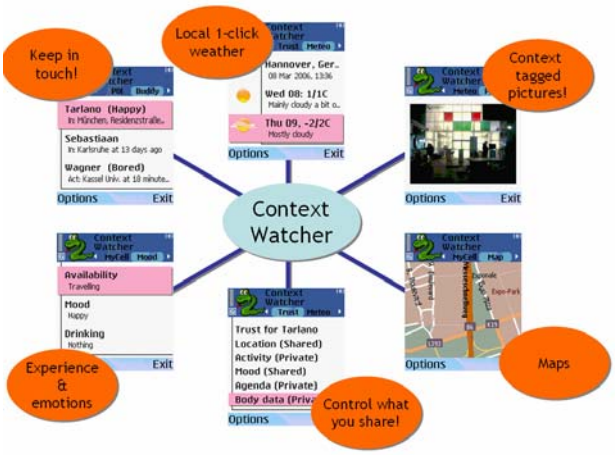

Figure 3: Main features of the Context Watcher

- Mood (based on user input)
- Activities and meetings (based on reasoning)
- Body data (based on heart and foot sensors)
- Weather (based on a location-inferred remote weather CP)
- Visual data (pictures enhanced with contextual data)

Context information, and location data in particular, can be used as automatic input parameter for services. Because the Context Watcher is aware of your location, it can send that data automatically to value-added services to obtain maps, nearby points of interest of some kind, or local weather information. The map and poi services require you to have a location information in terms of latitude and longitude (does not matter if this is GPS or cell-id based), whereas the weather service requires you to have location information in terms of city and country.

Among many appealing features, one can share context-enriched pictures with your friends. Results might look like these examples (http://www.flickr.com/photos/tags/mobilife) of some MobiLife people. Additionally, the Location Context Provider can write daily reports of your activities automatically and send these reports to your blog. An example is Johan Koolwaaij's blog (http://koolwaaij.blogspot.com/). The reports are generated overnight and are based on yesterday's data.

More detailed information and downloads on [7] or [6].

VI. Conclusion

The results as described in this paper have given an overview about how context-awareness has been addressed by the MobiLife project. From the Context Management Framework over the Context Provider description to the introduced sample applications the paper has shown the cycle from the conceptualisation of context-awareness to the usage of actual context information in real applications. Further research challenges have been addressed by the MobiLife project, for example work related to privacy and trust, an architectural reference model for all defined components and more. For more information refer to http://www.ist-mobilife.org.

Acknowledgment

This work has been performed in the framework of the IST project IST-2004-511607 MobiLife, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues.

References

[1] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," College of Computing, Georgia Institute of Technology, Tech. Rep. GIT-GVU-99-22, 1999

[2] R. Kernchen, P. Boda, K. Moessner, B. Mrohs, M. Boussard, G. Giuliani, "Multimodal user interfaces for context-aware mobile applications", 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2005, Berlin, Germany.

[3] A. Salden, R. Poortinga, M. Bouzid, J. Picault, O. Droegehorn, M. Sutterer, R. Kernchen, C. Räck, M. Radziszewski and P. Nurmi, "Contextual personalization of a mobile multimodal application", International Conference on Internet Computing (ICOMP), 2005, Las Vegas, USA.

[4] O. Coutand, O. Droegehorn, K. David, P. Nurmi, P. Floréen, R. Kernchen, S. Holtmanns, S. Campadello, T. Kanter, M. Martin, R. van Eijk, R. Guarneri, "Context-aware Group Management in Mobile Environments", IST Mobile and Wireless Communications Summit, 2005, Dresden, Germany.

[5] P. Floréen, M. Przybilski, P. Nurmi, J. Koolwaaij, A. Tarlano, M. Wagner, M. Luther, F. Bataille, M. Boussard, B. Mrohs, S. Lau "Towards a Context Management Framework for MobiLife" In proceedings of the IST Mobile & Communications Summit, Dresden, 19-23 June, 2005.

[6] MobiLife project, http://www.ist-mobilife.org (select 'Software' from the menu)

[7] J. Koolwaaij, "Context Watcher", http://www.lab.telin.nl/~koolwaaij/showcase/crf/cw.html