Distributed Context Management in a Mobility and Adaptation Enabling Middleware (MADAM)

Marius Mikalsen[1] Nearchos Paspallis[2] Jacqueline Floch[1]
Erlend Stav[1] George A. Papadopoulos[2] Akis Chimaris[2]

[1]SINTEF ICT
NO-7465 Trondheim
Norway
+47 73 59 30 00
{mariusm,jacf,sta}@sintef.no

[2]Department of Computer Science
University of Cyprus
P.O. Box 537 CY-1678 Nicosia, Cyprus
+357 22 89 26 84
{nearchos,george,cspgha}@cs.ucy.ac.cy

1 ABSTRACT

As computing devices are getting smaller, we tend to bring them everywhere. Consequently the operating conditions of the devices are constantly changing (e.g. changing user requirements, change in the system context and environment context). In order to be usable and dependable, applications and services need to self-adapt to changes in context. This work describes a context management approach for reducing the complexity of context aggregation and utilisation. The context manager is a core component in the MADAM (Mobility and ADaptation enAbling Middleware) project.

Categories and Subject Descriptors

D2.10 [**Software Engineering**] Design- *Representation.*D.2.11 [**Software Engineering**]: Software Architectures – *Data Abstraction, Information hiding.*

General Terms

Design, Performance, Reliability, Human Factors.

Keywords

Context, context awareness, adaptation-enabling architecture, middleware.

2 INTRODUCTION

Smaller computing devices, increasing computing power and proliferating mobile networks encourage users to bring their computers everywhere they go. This leads to diverse operating conditions, and in order to remain usable, systems should be able to self-adapt to the changing conditions [1]. Although the utility of adaptive and context aware systems have already been demonstrated by [2], the ubiquitous computing paradigm suggested by Weiser [3] is yet to come. One of the reasons for this is likely to be the development costs of such systems and the lack of generic tool support in the development process [4].

To encourage their uptake, cost-effective development of adaptive and context aware applications must be possible. This work describes a middleware architecture with particular focus on the context management part of the architecture that aims to provide tool support for the development of context aware applications.

SAC'06, April, 23-27, 2006, Dijon, France.

3 THE MADAM SERVICES

The adaptation middleware includes three main services:

- The Context manager which monitors changing user requirements and (system and environment) context for detection of relevant changes.
- The Adaptation Manager which reasons about the impact of the changes and decides about appropriate adaptations.
- The Configurator which reconfigures the component-based application to put the decided adaptations into effect.

4 CONTEXT

4.1 Theory of context

We have adopted Dey's widely used and broad definition of context [5]: *"Context is any information that can be used to characterize 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."*

This broad definition is applicable because MADAM intends to be platform and domain independent, and as such, cannot define context in a way that might exclude certain platforms or domains.

4.2 The MADAM context model

Building on the work described in [6], we suggest the context model depicted in Figure 1.

The context information is encompassed in context elements. The context elements can be composite (elements within elements), and they are associated to values. The value is the information available in the element. Every value instance encapsulates a data entity, of any of the following types: string, integer, long, float, double, or boolean. Values are associated with metadata that provide additional information about the value characteristics. Metadata information is essential when performing reasoning on context information.

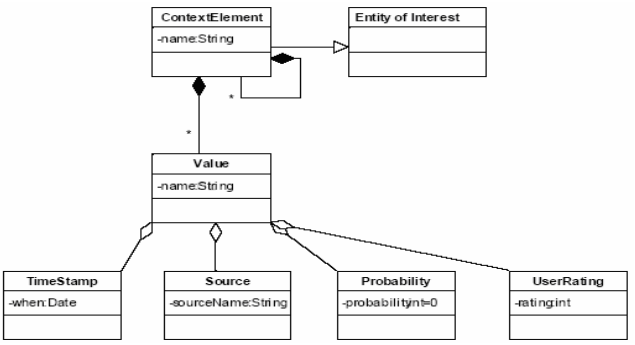

Figure 1 - The MADAM context model

5 CONTEXT MANAGER

In MADAM, context management is defined as:

- Providing a standardised way to represent context
- Providing standardised interfaces and methods to access context information from sources that monitor and report context information (context sources)
- Providing functionality to store and retrieve context (context querying)
- Providing reduction of context noise through proper context access mechanisms
- Ensuring consistency in context representations, by making sure that no context elements are duplicated
- Ensuring consistency with regards to context sensors, context reasoners, and context storages, by making sure that none are duplicated.

5.1 Context manager architecture

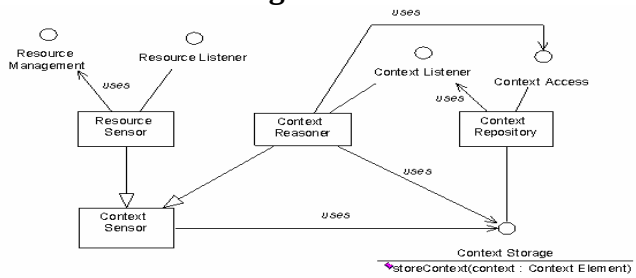

Figure 2 - The context manager architecture

The context manager considers entities that need context information to be context clients, while it considers entities providing context information to be context sources [7]. In MADAM, the primary context client is the adaptation manager, while context sources are typically context sensors and context reasoners. A context source can also be a context client (e.g. a context reasoner which requires context information from other sources in order to provide new context information).

Context Repository The context repository is the main entry point for clients to the context manager. The primary tasks of the context repository are to maintain a context model, register and notify listeners, give access to context elements, and keep registry of available components (sensors, reasoners and storage).

Context Sensors The context sensors are components which provide context information to the context repository (a type of context source). Sensors can for instance be wrappers around legacy code used for monitoring context changes, such as battery, memory, and network information.

Context Reasoners Context reasoners are specialisations of context sensors. The main difference from sensors is that context reasoners can produce one or more context elements using other context elements as input.

Context Storage Keeping track of historical context information is required in order to be able to determine trends in context data, such as user behaviour and network stability.

6 CONCLUSION

In this paper we have argued that mobile use and continuously changing operating conditions require that applications adapt accordingly. Standardized middleware tools targeting such mobile devices and applications, become increasingly important, as they can facilitate the development of reliable and usable software.

We have proposed a generic and platform independent context model and context management mechanism. By providing a necessary abstraction level and reducing context noise, a well functioning context managing mechanism is a prerequisite for a well functioning adaptation enabling middleware. We currently have a proof of concept implementation that provides valuable insight, and further research is conducted to validate and refine the context information model and middleware architecture and components.

7 ACKNOWLEDGEMENTS

This work was funded by the EU commission as part of the IST MADAM project (6th framework programme, contract number 4169).

8 REFERENCES

[1] Hallsteinsen, S., Stav, E., and Floch, J. Self-Adaptation for Everyday Systems. In Proceedings of ACM SIGSOFT Workshop on Self-Managed Systems(WOSS '04), Oct 31-Nov 1, 2004 Newport Beach, CA, USA

[2] Bristow, H. W., Baber, C., Cross, J. and Wooley, S. (2002) Evaluating contextual information for wearable computing. In Proceedings of the Sixth International Symposium on Wearable Computers, New York: IEEE Computer Society.

[3] Weiser, M.: Some computer science issues in ubiquitous computing. Communications of the ACM 36 (1993)

[4] Kofod-Petersen, A. and Mikalsen, M.(2005) Representing and Reasoning about Context in a Mobile Environment. Revue d'Intelligence Artificielle (RIA), vol 19, no. 3, pp 479-498.

[5] Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing 5 (2001) 4–7

[6] Göker, A., Myrhaug, H.I.: User context and personalisation. In: Workshop proceedings for the 6th European Conference on Case Based Reasoning (2002).

[7] Lei H., Sow D. M., Davis J. S. I., Banavar G., Ebling M. R., "The design and applications of a context service", *Mobile Computing and Communications Review*, 6(4): 44–55, 2002. ACM SIGMOBILE