A Coordination Protocol for Distributed Context Management Systems

J. Schneider, C. Mannweiler, A. Klein, H.D. Schotten
Chair for Wireless Communications and Navigation, University of Kaiserslautern, Germany
{schneider, mannweiler, aklein, schotten}@eit.uni-kl.de

***Abstract*—Context Data and Context Management Systems have become more important over the last years. Based on the heterogeneous context data landscape, several approaches for Context Management architectures have been proposed. Most are centralized and did not take care of scalability. In this paper, we will not develop a new Context Management Architecture; we rather introduce an extension of a centralized context management system to get a distributed one. Furthermore, we developed and evaluated a data transmission protocol called Distributed Context Management Protocol (DCMP) which is based on the standard UDP and TCP protocol. In this paper, a detailed description of the context management system extension will be given, and we give a short description of our DCM protocol.**

Keywords: *Context Management, Distributed Context Management Protocol*

I. INTRODUCTION

Context acquisition as well as the management of context data has become important in recent years. Today, context information is made available for a broad variety of applications. The Internet of Things topic addresses these demands for developing comprehensive context-aware systems and services. Due to the resulting on heterogeneity of available context data, comprehensive systems or architectures have to be developed. Today, many companies and research groups work on projects for such context-aware systems. However most of the currently available systems are not standardized. An example for such a system has been studied in the IST-FP6 project "MobiLife" [1]. Comparable to the MobiLife context mamagement architecture, Chen [2] introduced a Context Broker Architecture (CoBrA) which uses a Web Ontlogy Language OWL for modeling ontologies of context and for supporting context reasoning. A main abandoning is addressed by the Context Broker. It is responsible for handling the context data.

The EU Project C-Cast [3] used a simple centralized Context Management architecture which consists of a Context Broker (CxB), Context Consumers (CxC), and Context Providers (CxP). A detailed functional description of this architecture is given in [4]. Mannweiler et al [5] described a distributed Context Management platform which is based on distributed hash tables. Our goal is not to develop an entirely new context management platform; rather, as described in the abstract, we want to enhance an available one. We decided to use the context broker architecture which has been used in the C-Cast project [3], [4]. This architecture provides a simple and efficient way for handling context data. A weakness of this approach is the deficient scalability. Therefore we enriched this architecture to a distributed one and developed an appropriate transmission protocol.

This paper is structured as follows: Section II presents an overview about the requirements for a distributed context management system. In Section III, we present our extension of the centralized context broker architecture and the developed protocol, called **D**istributed **C**ontext **M**anagement **P**rotocol (DCMP). In Section IV, we give a conclusion and describe future work.

II. REQUIREMNETS FOR A DISTRIBUTED CONTEXT MANAGEMENT PLATFORM

In this chapter, we give an overview on our developed system as well as the **D**istributed **C**ontext **M**anagement **P**rotocol (DCMP) and **C**ontext **B**roker **N**etwork **H**andler (CBNH). The function of a DCMP is comparable to a DHCP[7][8] and the CBNH to a DNS[9][10].

To develop a distributed context management system, several requirements have to be addressed:

- Plug and Play: automatic configuration, connection and disconnection of entities
- Data handling: distribution of the data in a meaningful way between the CxBs (e.g. load balancing)
- Simple implementation: utilization of standardized data format and data transmission protocols

For further work, we examine the centralized context broker architecture depicted in Figure 1. This centralized Context Management architecture has been used in the C-Cast project [6].

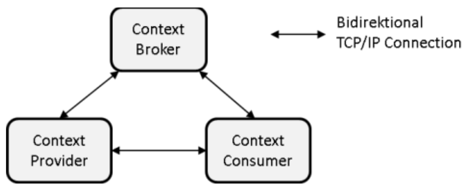

Figure 1: C-Cast Context Management architecture [6]

To enhance this architecture to a distributed one, we added a DCMP server on each CxB. This server is comparable to a DHCP server [7][8] with the difference that each client gets allocated to a suitable DCMP server. Additionally, the DCMP contains a Context Broker Network Handler. It is responsible

This work has been supported by the Federal Ministry of Education and Research of the Federal Republic of Germany (Foerderkennzeichen 01 BK 0808, GLab and 01 BU 1116, SolarMesh). The authors alone are responsible for the content of the paper.

for building up the CxB hierarchy. This is similar to the DNS structure, as described in later sections.

III. DCMP-Server and Context Broker Network Handler

For explaining system behavior, it is necessary to contemplate as a Context Provider / Consumer and a Context Broker.

A. Context Broker

During the startup of a CxB, it scans for other CxBs via a Broadcast request. If there are other active Brokers, the new CxB will send an "integrate" message to one of the active CxBs. Active brokers are aware of the system structure, and one of them sends all necessary information to the requesting CxB and all other involved parties. After this has been done successfully, the new CxB receives an "Ack" message.
For the case that no other CxB is active, the started one becomes "root" and is on top of the hierarchy. All CxPs and CxCs have to contact this CxB and therefore the system is comparable to a centralized architecture.
If an additional CxB starts up, is placed in the next level of the tree. In contrast to the original DNS specification [9][10] a new hierarchical organization need to be developed. For our application, we decided to use to setup the hierarchy into geographical regions. An example is listed in Table 1.

Table 1: Example for a hierarchical split

Top Level	**Second Level**	**Third Level**	**SubLevel**	**Host or Service**
Country	***Region***	***Position***	***Scope/Entity***	***Name***
Germany	RLP	Kaiserslautern	Environment/ Temperature	Fun18

Each context request by CxC will be handled recursively; hence it starts at the root node. To avoid an overload at the higher level nodes (first or second level) a caching function is added. Each CxB hosts a cache register. This register stores all requested CxP addresses with a timestamp. Each data request from a CxC is represented in a XML format, i.e. each CxB comprises a request resolver. This resolver is located in the Context Broker Network Handler and translates XML requests into an appropriate data format. Additionally, it is responsible for a recursive data requests. The hierarchical organization is done by the CBNH.

B. Context Provider and Context Consumer

Each CxP or CxC which enters the architecture needs to be register into the system. To provide or consume data, it is necessary to assign an appropriate CxB to these entities. We decided to integrate a DCMP server, which is responsible for handling this issue. If a CxP or a CxC wants to register, it sends a broadcast message on a defined port. These servers are listening on this port and repay this request with a corresponding message. Figure 2 shows the timeline message diagram between client and server. There is only one difference between an integration of a CxP and a CxC. For a CxP, it is necessary to choose an appropriate CxB to ensure the "data handling" requirement (compare chapter II), which has no effect for a CxC. Consequently, "Client register request" message has to include additional information about the entity. Each announcement is valid until the timestamp has not been expired. During this time "renew timestamp" message has to be send to the CxB.

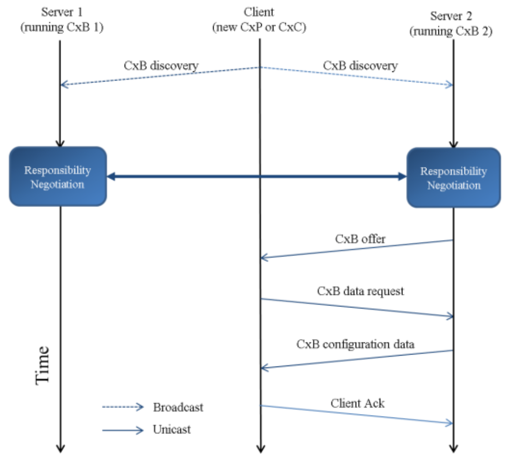

Figure 2: Timeline message diagram between client and server

If the system has been started successfully, all data transmissions act as described in [2].

IV. Conclusions and Future Work

With the extension of the centralized Context Architecture, we enabled the possibility to deploy distributed context management architectures. For the "Plug and Play" requirement, we included a DCMP server and a CBNH in each Context Broker. The DCMP server is comparable to a DHCP server and provides all necessary data to a CxP or CxC. The CBNH is similar to DNS and addresses the organization of the hierarchical CxB structure and takes care of the "Data handling" requirement. With the relation to the DNS service and the DHCP concept, we are able to provide a solution for a distributed context management architecture, which uses standardized TCP and UDP connections.

Future work includes a detailed research of appropriate strategies for data handling and distribution, hierarchical structuring. Additionally, use cases related to the Internet of Things scenario needs to be identified for ensuring the scalability of the system.

References

[1] Floren, P., Przybilski, M., Nurmi, P., Koolwaaij, J., Tarlano, A., Wagner, M.,Luther, M., Bataille, F., Boussard, M., Mrohs, B., Lau, S.: "Towards a Context Management Framework for MobiLife", *14th IST Mobile and Communications Summit*, Dresden (2005)

[2] Chen, H., Finin, T., & Joshi, A. (n.d.). "Semantic Web in a Pervasive Context-Aware Architecture", *Artificial Intelligence in Mobile System 2003, Volume 28*, 2003

[3] Project Context Casting (C-CAST), http://www.ict-ccast.eu/

[4] J. Schneider, A. Klein, C. Mannweiler, H. D. Schotten, "Environmental Context Detection for Context-Aware Systems", *ICaST - ICST's Global Community Magazine*, February 2010

[5] C. Mannweiler, B. Amann, J. Schneider, A. Klein, H. D. Schotten, "A Distributed Context Service Architecture for Heterogeneous Radio Environments", *ITG-Fachbericht der 15. ITG Fachtagung Mobilkommunikation, Osnabrück, Germany*, May 2010

[6] C-Cast Deliverable D7, "Requirements and concepts for context detection and context-aware multiparty transport", October 2008

[7] R. Droms, "Automated configuration of TCP/IP with DHCP", *IEEE Internet Computing*, July-August 1999

[8] R. Droms, "Dynamic Host Configuration Protocol", *Network Working Group*, http://datatracker.ietf.org/doc/rfc2131/, March 1997

[9] P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", Network Working Group, http://tools.ietf.org/html/rfc1035, December 1987

[10] P. Mockapetris, "DOMAIN NAMES - CONCEPTS AND FACILITIES", Network Working Group, http://tools.ietf.org/html/rfc1034, November 1987