Generic Context Acquisition and Management Framework

Paulo Costa, Luís Botelho

“We, the Body and the Mind” Research Lab ADETTI / ISCTE, Avenida das Forças
Armadas 1600-082 Lisboa Portugal
[paulo.costa, luis.botelho]@we-b-mind.org

Abstract. This paper presents a framework that enables context aware computation to the service domain. The use of context aware computing in service oriented programs allows users and applications to communicate in a simpler way making these services more appellative. This framework supports an approach for ubiquitous computing paradigm to service oriented programs such as web-services. The principal of this paradigm is to enhance usability of applications by letting them adapt to conditions that directly affect their operations. The architecture described in this paper is presents a solution for a generic context aware system.

1 Introduction

The capability of passing an idea to each other and naturally reacting to it is something that every person is capable of. When two people communicate, a great amount of implicit information is never directly transmitted between them. This implicit information is called context and it allows the participants to improve their interaction, providing a shared background in which the meaning of the communicative act becomes grounded in their situation. This is done without overloading the information explicitly conveyed.

These benefits can then be passed to the service domain through the use of context and context aware applications. Sensors and sensor networks are used to expand application communication with the environment [4]. To simplify the communication between these sensors and the applications this paper proposes a generic and easy to use context system architecture. This system uses an infrastructure approach [5] allowing every types of applications, including web-services, to access context information.

Services and applications can directly access this framework to query the system for context, register for context events or supply context information to the context system. This information is stored in repositories. This work has been supported in part by the European Commission under the project grant FP6-IST-511632-CASCOM

This paper illustrates in first place a detailed description of the architecture (section 2). Section 3 describes a prototype implementation of this architecture. Section 4 compares this proposal with related work. Finally, section 5 presents conclusions and future work.

2 System Architecture

The proposed systems' architecture is presented in figure 1. This architecture is divided in two levels. The application level, identified by the context interpreter, defines application specific modules that will be implemented by the applications that will use this subsystem. The generic level is responsible for the capture of context information from the sensors and its representation in a uniform way.

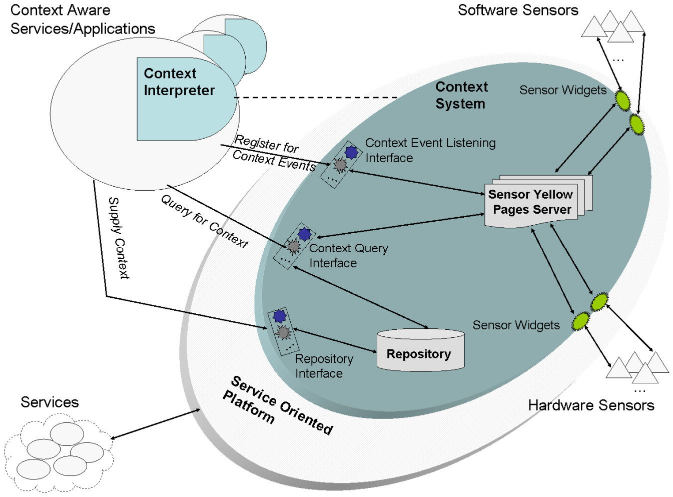

Fig. 1. Context system architecture

The generic level possesses several interfaces for context query and acquisition, event listening, context storage and sensor communication. The application specific level has a set of application specific components that interprets aggregates and reasons about context.

Context aware applications and services can directly access the context system trough its interfaces. The context system can also be used inside other platforms. In this case a service oriented platform is used. With the aid this platform other services that are not context aware can become so, as it will be described in section 3.

2.1 The Generic Level

In the generic layer, the repository interface provides the tools to receive and store the context provided by the applications. The gathered information is then stored in the repository for future searches. Sensor history information can also be stored on application requirement. Since this information consumes a great amount of space, it is only stored during a fraction of time specified by the application that required it.

The Context Query interface provides interprets the queries made by the applications and provides the context information required. This interface is responsible for defining a context query language and presenting a high level mechanism for posing queries. It uses the yellow pages server to find the context sources required for the query and captures the context information from the acquired context.

The Context Events Listening interface listens for context alterations and warns the registered applications when the context changes. This functionality is responsible for defining the available event listeners according to the existing sensors, specifying the application requests and their association with the event listeners and managing application subscription.

The sensor widgets provide a communication interface between the sensors and the information gatherers. These widgets also provide registration mechanisms in the yellow pages service. The widget approach allows several different sensors to interact with the subsystem making the sensorial interfaces compatible with the platform specifications [4].

The generic design of the proposed system allows it to adapt to any kind of context information. A context model of the surrounding environment describing the context of each element must be made before this system can be used [2]. The context system and all its client applications must be aware of the context model.

2.2 The Application Level

In the application layer, the Context Interpreter models the gathered context information in a way that can be understood by the application. The Context Interpreter uses the aggregation mechanism to join sensor information and the context reasoning mechanism to infer new context. Due to the specificity of this module it uses application-defined ontology.

The aggregation mechanism is used when it is necessary to merge correlated context information. Its main function is to define merging mechanisms and an ontology that supports the merging. The reasoning mechanism is used when is necessary to infer new context. It defines reasoning mechanisms that produce new context from collected context information, enriching it with new information.

3 Prototype

To prove the effectiveness of the presented framework, an application scenario has been defined. This scenario is part of the emergency assistance scenario defined in the CASCOM project [7]. This scenario defines a medical assistance platform based on services that use several components to find, compose and execute these services. Its main propose is the aggregation of different services with the aid of context information, in order to obtain a user determined objective. The interaction between these can be seen on figure 2.

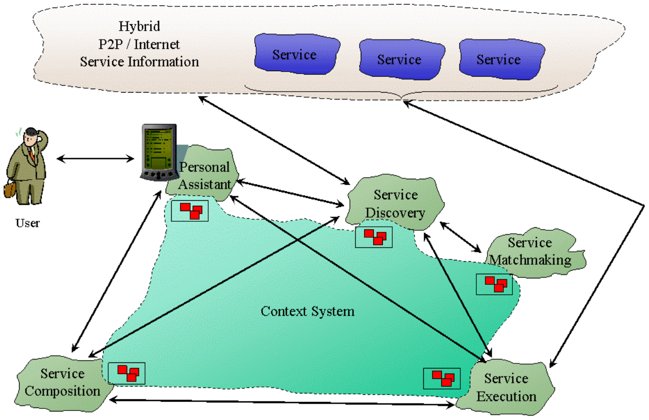

Fig. 2. Interaction between application components

Each one of these components makes use of context information to improve general performance. The platform will work on top of a peer-to-peer based network, where several medical related services can be found.

As it can be seen in figure 2, services do not need to be context aware since all context related processing is done while searching, composing or executing them on this prototype platform. In some aspects this prototype acts like the middle layer proposed by Ritchie [8] since services are not aware of the use of context.

The personal assistant exists to facilitate user communication with the application. With it the user can find the medical services that suit his needs. With this prototype users can then find medical services that are suited to their situation, wherever they are. Other services can also be used by this platform, making this a practical solution to solve problems that involve interaction between several different entities. The use of context awareness in such cases is a great advantage since it narrows down the number of possible solutions to the ones that fit exactly the situation profile.

4 Related Work

Context awareness has recently gained more attention in the research community, and currently there are signals indicating that context-aware services could be significant business. Several approaches tried to simplify the introduction on context awareness on service oriented computing. Approaches like the CoBra architecture [1] present solutions for agent based networks. This architecture is suitable for intelligent services like intelligent houses or cars. It is based on a central component that introduces a general vision of the context to the other agents.

The WASP platform [3] is a web services based context aware service platform. This platform provides a generic architecture that supports the execution of adaptive context-aware applications. Like the architecture presented in this paper it possesses mechanisms and interfaces that shield services from the complexity introduced by handling contextual information.

In order to introduce context awareness without changing the services, Richie [8] proposes the use of a middle layer between the application that uses the service and the service itself. This layer will then provide context awareness to the selected services without this implying changing the service itself. Several other architectures provide context solutions for general applications that can be adapted to the services, like the GAIA middleware [6] or the Context Fabric [5].

5 Conclusions and Future Work

In this paper a generic context architecture that is adaptable for all kinds of applications has been presented. The main contribution of this proposal consists on the support for the introduction of ubiquitous computing paradigm into applications. By removing the burden of obtaining information from sensors, locating and registering sensorial devices and managing stored contextual information from the applications themselves, application developers can focus only on application specific problems and on what to do with the information provided by the context system.

By combining this system with other platforms solutions like the presented prototype the use of context awareness can be expanded to other domains. With the aid of this service oriented platform normal services can become context aware with no alteration necessary. As long as services are compatible with the platform communication protocols the platform can use context information to find, compose and execute the services required by the user and that are adapted to the situation.

With the aid of the application level modules, applications and services can also enhance the way they handle context. Raising context abstraction level and obtaining more detailed context information are the improvements brought by this context system plug-in.

Test results can not be yet displayed since the prototype application is still on development stage while this paper is being written. As far as can be seen, the major difficulties that can emerge are related to context information itself. The context model must represent in a reliable way the application environment or else the results can be erroneous.

A great amount of effort is still to be taken on the development of the application specific plug-ins. These plug-ins are not going to be implemented in this version of the prototype due to the simplicity of the application scenario that does not imply the use of context interpretation mechanisms. More complex implementations may require it and due to it the development of these plug-ins is indispensable.

References

1. Chen, H., Finin, T., Joshi, A.: An Intelligent Broker for Context-Aware Systems. In Adjunct Proceedings of Ubicomp, Seattle, Washington, USA (2003)

2. Christopoulou, E., Goumopoulos, C., Zaharakis, I., Kameas, A.: An Ontology-based Conceptual Model for Composing Context-Aware Applications. Research Academic Computer Technology Institute (2004)
3. Costa1, P., Filho, J., van Sinderen, M.: Architectural Requirements for Building Context-Aware Services Platforms. IFIP workshop on Next Generation Networks, Balatonfured, Hungary (2003)
4. Dey, A. K., Abowd, G., Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction (2001)
5. Hong, J., Landay, A.: An Infraestruscture Approach to Context-Aware Computing, Human-Computer Interaction (2001)
6. Kon, F., Hess, C. K., Román, M., Campbell, R. H.: A Flexible, Interoperable Framework for Active Spaces. OOPSLA'2000 Workshop on Pervasive Computing, Minneapolis, MN, (2000)
7. Möller, T., Schirmer, P., Schuldt, H., Kinnunen, A., Stark, C., Vogel, R.: Emergency Assistence – Use Case Scenarios. FP6-IST-511632-CASCOM project. TeliaSonera (2004)
8. Ritchie, M. Pre & Post Processing for Service Based Context-Awareness. Technical Report Equator-02-023, University of Glasgow / Department of Computing Science (2002)