A QUALITY OF CONTEXT DRIVEN APPROACH FOR THE SELECTION OF CONTEXT SERVICES

Elarbi Badidi[1] and Larbi Esmahi[2]
[1]*College of Information Technology, United Arab Emirates University, PO.Box. 17551, Al-Ain, United Arab Emirates*

[2]*School for Computing & Information Systems, Athabasca University*
1 University Drive Athabasca, Alberta, T9S 3A3, Canada

Keywords: Mobile Workers, Publish-subscribe, Quality-of-context, Context-aware Web Services, Service Selection.

Abstract: As wireless networks and mobile devices are becoming ubiquitous, mobile users are increasingly requiring access to application services that can adapt to their context as they move to new locations, for example, in their corporate or partners' networks. The quality-of-context information (QoC) used by these application services is a determinant factor in the adaptation process. As application services typically receive context data from several context services, the selection of suitable context services is of paramount importance in providing mobile users with tailored services. In this paper, we describe our proposed framework for context management and our proposed QoC-based algorithm for the selection of context services. The algorithm takes into account the QoC requirements of application services for each context information to which they subscribe with the Context Broker, on which the framework is relying for context management.

1 INTRODUCTION

As wireless networks and mobile devices are becoming ubiquitous, there is an unprecedented rise in the number of mobile workers who are using a variety of modern handheld devices such as PDAs and SmartPhones to consume online services.
With this proliferation of mobile devices, wireless business applications, i.e. messaging and voice services, and healthcare services are more and more developed and deployed using Web services. Thus, using the Service Oriented Computing paradigm in mobile environments considerably enlarges the range of accessible business applications and enables delivering integrated services across wireless networks.

Mobile workers are increasingly requiring retaining access to services that are similar to their corporate services as they move to new locations in a very straightforward manner without having to configure their working environment explicitly. Therefore, business services should be context-aware to deal with the changing environment of the user. Several definitions of the notion of context have been provided in the literature. According to Dey's definition (Dey (2001)), the amount of information that can be categorized as context information is extremely broad. Location and time are the most widely used context parameters by applications. Wireless devices such as environmental sensors, radio frequency identification (RFID) tags send raw context information such as location, presence and other status information across the network. Specialized services, that we call *context services*, capture, store, analyze and aggregate data to provide high-level context information to application services as needed. Indicators that can essentially be measured and captured include temperature, humidity, pressure, whether the mobile worker is in motion, and many more.

Being context-aware allows application services to make inferences about the current situation of the user using context information obtained from various sources such as surrounding sensors, and GPS receivers, and hence adapt their behavior to the prevailing context of the user. By fusing acquired data and inferring on it, these applications can deduce, for instance, the position or the current user activity such as "user is in a shopping mall" or "user is in a meeting". Such Context-aware application services are being deployed in various industries

Badidi E. and Esmahi L. (2010).
A QUALITY OF CONTEXT DRIVEN APPROACH FOR THE SELECTION OF CONTEXT SERVICES.
In *Proceedings of the International Conference on e-Business*, pages 89-94
DOI: 10.5220/0003031200890094

such as healthcare, airline transportation, manufacturing, and retail. In the last few years, Context-aware Web services have emerged as a promising technology for building innovative and interoperable context-aware applications (Truong et al. (2009)).

One of the main problems being faced in the area of ubiquitous computing is handling and distributing context efficiently to enhance personalized service delivery to mobile users. The context sources as well as the consumer services of context information are very often physically distributed. For instance, the context sources providing information about the current temperature may be far from the applications that need to adapt their services to the prevailing weather conditions. Furthermore, it is likely that these context sources provide the same context information but with different QoC (Buchholz et al. (2003); Kamran et al. (2008)). Context-awareness raises new challenges like aggregation of context information in a structured format, discovery and selection of suitable context sources.

In this paper, we describe our proposed framework for context management and our proposed QoC-based algorithm for the selection of context services by application services, so that they can adapt their services to the user context. The algorithm takes into account the QoC requirements of the application services for each context information to which they subscribe with the *Context Broker*. The framework is relying on the *Context Broker* for context management. Selection of context-aware application services by users or intermediary entities is not the subject of this work as it has been investigated by several research works (Yu et al. (2009); Kirsch-Pinheiro et al. (2008)). To the best of our knowledge, selection of context services has not been investigated before.

The *Context Broker* implements a *topic-based publish/subscribe* model in which application services subscribe to context information they are interested in, and context services publish their context information. Context Brokers have been used also in the following works (Chen et al. (2003); Bonino da Silva Santos et al. (2007)).

2 BACKGROUND

2.1 Quality-of-Context

Context information is characterized by some properties referred in literature as quality-of-context (QoC) indicators. Buchholz et al. (Buchholz et al. (2003)) have defined the QoC as: *"Quality of Context (QoC) is any information that describes the quality of information that is used as context information. Thus, QoC refers to information and not to the process nor the hardware component that possibly provide the information."*

Buchholz et al. (2003)) and Kamran et al. (2008) have identified the following QoC indicators: *precision*, *freshness*, *temporal resolution*, *spatial resolution*, and *probability of correctness*.

Precision. This indicator represents the granularity with which context information describes a real world situation.

Freshness. The time that elapses between the determination of context information and its delivery to a requester.

Spatial Resolution. The precision with which the physical area, to which an instance of context information is applicable, is expressed.

Temporal Resolution. The period of time during which a single instance of context information is applicable.

Probability of Correctness. This indicator represents the probability that a piece of context information is correct.

Several competing *context services* may provide the same context information (Buchholz et al. (2003)). Therefore, potential context consumers should be able to select context services on the basis of the QoC they can assure.

2.2 Context Services

A context service typically provides infrastructure support for collection, management, and dissemination of context information concerning a number of subjects. Subjects may be users, objects such as handheld devices and equipments, or the environment of users. The context service acquires context information from various context sources. Sources are usually third parties that collect and provide context information. For example, consider the "temperature" at the current location of the mobile user. This information may be obtained directly from the mobile device of the user. It can also be obtained from a local weather station. Alternatively, it may be obtained from weather TV channels providing weather information nation-wide.

Several research works have investigated the design and the implementation of context services. Shmidt et al. designed and implemented a generic context service with a modular architecture that

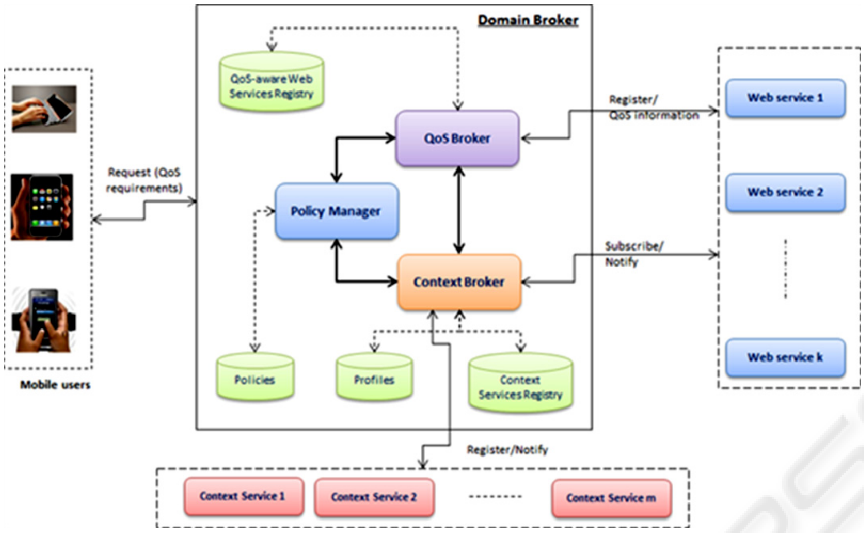

Figure 1: Framework Components.

allows for context collection, discovery and monitoring (Shmidt et al. (2009)). This context service provides a Web service interface that allows its integration in heterogeneous environments. The implementation uses OWL to describe context information and SPARQL to query and monitor context information.

Lei et al. described the design issues and the implementation of a middleware infrastructure for context collection and dissemination (Lei et al. (2002)). They realize this middleware infrastructure as a context service. To allow for wide deployment of the context service, this work has addressed the following issues: Extensibility of the context service architecture by supporting heterogeneous context sources, integrated support for privacy, and quality of context information support. Coronato et al. proposed a semantic context service that relies on semantic Web technologies to support smart offices (Coronato et al. (2006)). It uses ontologies and rules to infer high-level context information, such as lighting and sound level, from low-level raw information acquired from context sources.

3 FRAMEWORK OVERVIEW

As illustrated in Figure 1, our proposed framework is relying on a *Domain Broker*, which mediates between mobile users and application services at the visited site. The *Domain Broker* components are the *QoS Broker* and the *Context Broker*. Another key component of the framework is the *Policy Manager*, which is responsible for managing and maintaining authentication and authorization policies, as well as polices for monitoring services and their quality of service. The *Domain Broker* components collaborate to deliver personalized context-aware services to mobile users with various devices across interacting sites. The *QoS Broker* is in charge of managing the QoS of Application Web services and submitting the mobile user requests to suitable ones. The *Context Broker* is in charge of managing context information and user profile and preferences. QoS Management operations (QoS specification, monitoring, service level agreement (SLA) negotiation) performed by the *Qos Broker* components are described in our previous work (Badidi et al. (2009)). This paper extends the proposed architecture by considering the context dimension so that applications services are context-aware.

When applications are context-aware, they can adapt their behavior and offer to the user contextually relevant information. Knowledge of the user context allows anticipating the user service and information needs. Context management is achieved in our framework by using a *Context Broker* and by adopting a *topic-based Publish/subscribe* messaging model, a one-to-many pattern of asynchronous message distribution based on registration of interest. Publishers label each message with the name of a topic ("publish") rather than addressing it directly to subscribers. The message system then sends the message to all eligible recipients that expressed their interest in receiving messages on that topic ("subscribe"). The publish/subscribe model is a loosely coupled architecture in which senders often do not need to know who their potential subscribers are, and the subscribers do not need to know who generates the information.

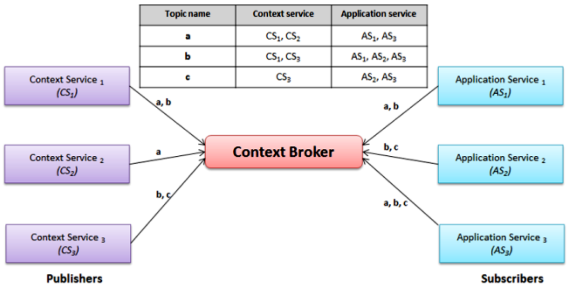

Figure 2: Topic-based publish/subscribe system.

It is increasingly being used in a service oriented architecture context. A Web service disseminates information to a number of other Web services, without the need to have prior knowledge of these other services.

The *Context Broker* is a mediator Web service that decouples context consumers from context services. It is in charge of handling subscriptions of application Web services in which they express their interest to receive context information, and registrations of context services. Once context services are registered with *the Context Broker*, they publish context information and application Web services are then notified by the *Context Broker* about the new context information. Figure 2 illustrates our topic-based publish/subscribe system in which context services are the publishers and the application services are the subscribers. Context information -- such as location, temperature, and user activity -- represents the topics of the system.

4 QoC-BASED CONTEXT SERVICES SELECTION

As we have mentioned earlier, context information may be delivered with different QoC by various context services. Therefore, the *Context Broker* is in charge of selecting appropriate context services to deliver context information to which application services subscribe. Context information may be delivered to the same application service by several context services. Each one may deliver a piece of context information (a topic) that the application service requires to adapt its behavior to the current context.

In the following, we describe the context services selection algorithm for a given application service.

Let $T = \{t_1, t_2, \dots, t_k\}$ be the list of context information (topics) to which an application service has subscribed by showing its interest in receiving such context information.

Let $C = \{CS_1, CS_2, \dots, CS_l\}$ be the list of context services registered with the *Context Broker*. Two context services may provide different context information; each one specializes in offering particular context information. One service, for example, may offer location information while another service may offer only temperature information, and a third one may offer both of them.

These services typically provide context information with different QoC. We assume that QoC indicators are in normalized form with values between 0 and 1. A value of 1 means highest quality and 0 means lowest quality. For example for the *freshness* quality indicator, 1 means that context sources have sensed the information in the last minute, and 0 means that they have sensed it in the last 10 minutes.

When subscribing to context information, an application service specifies the min values of the normalized QoC indicators that it can tolerate. For instance, the application service may subscribe to the *location* information may require a min value of 75% for the *freshness* indicator and 95% for the *probability of correctness* indicator. Let $P = \{P_1, P_2, \dots, P_m\}$ be the list of QoC indicators (parameters) considered in the system.

The minimum quality requirements that the application service tolerates for a given context information (topic) t_j, with $1 \le j \le k$, are expressed by the following vector:

$$M_j = \{min_{1,j}, min_{2,j}, \dots, min_{m,j}\}$$

$0 \le min_{i,j} \le 1$, with $1 \le j \le k$ and m is the cardinality of P.

Therefore, the whole quality-of-context requirements of the application service for all its subscribed topics and all QoC indicators considered in the system can be expressed by the following matrix:

$$Q_a = \begin{array}{c} \\ t_1 \\ \vdots \\ t_2 \\ \cdots \\ \cdots \\ t_k \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_2 \dots\dots\dots & & P_m \end{array} \\ \begin{bmatrix} min_{1,1} & min_{2,1} \cdots & \cdots & min_{m,1} \\ \vdots & & & \vdots \\ min_{1,2} & min_{2,2} \cdots & \cdots & min_{m,2} \\ & & & \\ \vdots & & & \\ min_{1,k} & min_{2,k} \cdots & \cdots & min_{m,k} \end{bmatrix} \end{array}$$

The goal of the selection algorithm is to find for each topic t_j, to which the application service subscribed, a suitable context service from the set C

that can satisfy the minimum quality requirements of the application service.

The QoC offer of a context service $\mathrm{CS_r}$ is expressed by the following matrix:

$$Q_r = \begin{array}{c} \\ t_1 \\ \\ t_2 \\ \dots \\ \dots \\ t_k \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_2 \dots\dots\dots\dots & & P_m \end{array} \\ \begin{bmatrix} q_{1,1}^r & q_{2,1}^r \cdots & \cdots & q_{m,1}^r \\ \vdots & & & \vdots \\ q_{1,2}^r & q_{2,2}^r \cdots & \cdots & q_{m,2}^r \\ \vdots & & & \\ q_{1,k}^r & q_{2,k}^r \cdots & \cdots & q_{m,k}^r \end{bmatrix} \end{array}$$

$\mathrm{CS_r}$ is suitable for a topic t_j if the following condition is satified:

$0 \le min_{i,j} \le q_{i,j}^r \le 1$ for $1 \le i \le m$ and $1 \le j \le n$ (1)

We define the distance of the QoC offer of $\mathrm{CS_r}$ from the application service required QoC for each quality indicator as:

$d_{i,j}^r = q_{i,j}^r - min_{i,j}$ for $1 \le j \le k$ and $1 \le i \le m$

Therefore, we can consider the distance matrix $Dist_r$ for the QoC offer $\mathrm{CS_r}$.

$$Dist_r = \begin{array}{c} \\ t_1 \\ \\ t_2 \\ \dots \\ \dots \\ t_k \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_2 \dots\dots\dots\dots & & P_m \end{array} \\ \begin{bmatrix} d_{1,1}^r & d_{2,1}^r \cdots & \cdots & d_{m,1}^r \\ \vdots & & & \vdots \\ d_{1,2}^r & d_{2,2}^r \cdots & \cdots & d_{m,2}^r \\ \vdots & & & \\ d_{1,k}^r & d_{2,k}^r \cdots & \cdots & d_{m,k}^r \end{bmatrix} \end{array}$$

Using the distance matrix, we can say that the application service requirement can be satisfied for a given topic t_j by the context service $\mathrm{CS_r}$ if the corresponding row in the above matrix has values greater than or equal to zero. Therefore, we can discard from that matrix the rows having negative values. We call the resulting matrix $PDist_r$ (all rows have positive values).

The Euclidian distance of $\mathrm{CS_r}$ offer from the application service QoC requirement for topic t_j is:

$$d_j^r = \sqrt{\textstyle\sum_{1 \le i \le m} \left(d_{i,j}^r\right)^2} \quad \text{with } 1 \le j \le k \qquad (2)$$

The highest value of d_j^r corresponds to the best QoC offer that can fulfill the QoC requirements of the application service for the topic t_j.

The most suitable context service for topic t_j, that we call here $target_j$, will be the one that maximizes the above Euclidian distance, that is :

$$target_j \leftarrow \max_{1 \le r \le l}\left(\sqrt{\textstyle\sum_{1 \le i \le m}\left(d_{i,j}^r\right)^2}\right) \qquad (3)$$

$$target_j \leftarrow \max_{1 \le r \le l}\left(\sqrt{\textstyle\sum_{1 \le i \le m}\left(q_{i,j}^r - min_{i,j}\right)^2}\right) \qquad (4)$$

In (4) we have assumed that the application service gives the same weight to all QoC indicators. This is not always the case as the application service may set relative weights for the QoC indicators. The application service may even set weights for each topic to which it subscribed. For example, for the *location* topic, more weight may be given to the *spatial resolution* indicator than to the *probability of correctness* indicator. For the *time of the day* topic, more weight may be given, for example, to the *precision* indicator than to the other QoC indicators. Therefore the weights for a given topic t_j, can be expressed by the following vector:

$$W_j = \{w_{1,j}, w_{2,j}, \dots, w_{m,j}\}$$

Where $0 \le w_{i,j} \le 1$ and $\sum_{1 \le i \le m} w_{i,j} = 1$

Given these weights, the most suitable context service for the topic t_j, $target_j$, will be the one that satisfies the condition (1) and which maximizes the sum of quality offers for all QoC indicators:

$$target_j \leftarrow \max_{1 \le r \le l}\left(\sqrt{\textstyle\sum_{1 \le i \le m}\left(w_{ij}\left(q_{i,j}^r - min_{i,j}\right)\right)^2}\right) \qquad (5)$$

If no context service satisfies the application service QoC requirements for a given topic, then the Context Broker may ask the application service to lower its QoC expectations.

The key idea of the above QoC-based context service selection algorithm is to find the most suitable context service with regard to the QoC requirements of a given application service for each context information (topic) to which the application service has subscribed.

5 PROOF OF CONCEPT

As a proof of concept, we describe in this section a scenario of how the QoC selection algorithm works.

Assume that the topics to which an application service has subscribed are in order of *location*, *time of day*, *activity*, and *temperature*.

Assume that the QoC indicators considered in the system are in order of *freshness*, *probability of correctness*, *temporal resolution*, and *spatial resolution*. The normalized minimum QoC requirements of the application service are given in Table 1.

Assume that four context services have registered with the context broker, CS_1, CS_2, CS_3, and CS_4. In the following we consider only the *location*; the same process applies to the other topics. Table 2 describes The QoC offer of the four Context services for the *location* context information (topic).

Table 1: Minimum QoC requirements of the application service.

	freshness	Probability correctness	Temporal resolution	Spatial resolution
location	0.95	0.75	0.65	0.75
time	0.80	0.85	0.80	0.65
activity	0.65	0.75	0.65	0.50
temperature	0.85	0.90	0.50	0.90

Table 2: QoC offers of four Context service for the *location* topic.

	freshness	Probability correctness	Temporal resolution	Spatial resolution
CS_1	0.97	0.80	0.75	0.85
CS_2	0.80	0.70	0.80	0.65
CS_3	0.98	0.78	0.65	0.80
CS_4	0.95	0.80	0.70	0.80

In this scenario, we consider that the application service assigns the same weight to all QoC indicators. By computing the distance matrix for QoC offers for the *location* topic and the Euclidian distance for each context service, we get the following ranking of the context services from highest offer to lowest:

CS_1 (0.1513), CS_4 (0.0866), CS_3 (0.0.0656)

The QoC offer of CS_2 does not meet the minimum QoC requirements of the application service.

6 CONCLUSIONS

Context services are increasingly used as an intermediary between context-aware application services and context sources. They provide infrastructure support for collection, management, and dissemination of context information concerning a number of subjects. The adaptation of services to the context of users requires the acquisition of high-quality information from context services. Therefore, the selection of suitable context services is becoming a pressing issue. In this paper, we have presented our proposed framework for context management and our proposed QoC-based algorithm for the selection of context services. The algorithm takes into account the QoC requirements of the application services for each context information to which they subscribe with the *Context Broker*, on which the framework is relying for context management.

REFERENCES

Badidi, E., Esmahi, L., Serhani, M. A., and Ikbal, T., 2009. A QoS-Broker Based Framework for Personalized Web Services Provisioning to Mobile Users, In *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS'09)*, Milan, Italy, July 2009, 147-152.

Bonino da Silva Santos, L. O., van Wijnen, R. P., and Vink, P., 2007. A service-oriented middleware for context-aware applications. In *Proc. of the 5th international Workshop on Middleware For Pervasive and Ad-Hoc Computing at Middleware 2007*, pp.37-42.

Buchholz, T., Kpper, A., Schiffers, M., 2003. Quality of context: What it is and why we need it. In *Proc. of the 10th International Workshop of the HP OpenView University association (HPOVUA).*

Chen, H., Finin, T., and Joshi, A., 2003. An ontology for context-aware pervasive computing environments. *Knowledge Engineering Review* 18: pp.197-207 Cambridge University Press.

Coronato, A., De Pietro, G., and Esposito, M., 2006. A Semantic Context Service for Smart Offices. In *Proc. of the 2006 international Conference on Hybrid information Technology - Volume 02*, pp.391-399.

Dey, A.K., 2001. Understanding and Using Context. *Journal of Pervasive and Ubiquitous Computing*, 5(1), pp. 4–7, 2001.

Kirsch-Pinheiro, M., Vanrompay, Y., and Berbers, Y., 2008. Context-Aware Service Selection Using Graph Matching. In *Proc. of the 2nd Workshop on Non Functional Properties and Service Level Agreements in Service Oriented Computing at ECOWS 2008.*

Lei, H., Sow, D. M., Davis, J. S., Banavar, G., and Ebling, M.R., 2002. The design and applications of a context service. *SIGMOBILE Mob. Comput. Commun. Rev.* 6, 4 (Oct. 2002), pp.45-55.

Schmidt, H., Flerlage, F., Hauck, F.J., 2009. A generic context service for ubiquitous environments. In *Proc. of the IEEE International Conference on Pervasive Computing and Communications (PERCOM 2009)*, pp.1-6.

Sheikh, K., Wegdam, M., and van Sinderen, M., 2008. Quality-of-Context and its use for Protecting Privacy in Context Aware Systems. *Journal of software*, 3(3), pp.83-93, March 2008

Truong, H.L., and Dustdar, S., 2009. A Survey on Context-aware Web Service Systems. *International Journal of Web Information Systems*, 5(1), pp.5-31, Emerald, 2009.

Yu, H.Q., and Reiff-Marganiec, S., 2009. Automated Context-aware Service Selection for Collaborative Systems. In *Proc. van Eck, J. Gordijn, R. Wieringa (eds.): Advanced Information Systems Engineering 2009 (CAiSE'09)*, LNCS 5565, pp. 261-274.