Service Delivery in Context Aware Environments: Lookup and Access Control Issues

Maria Riaz, Saad Liaquat Kiani, Sungyoung Lee, Sang-Man Han, Young-Koo Lee
Real Time & Multimedia Lab, Kyung Hee University, Giheung, Yongin, Gyeonggi-Do, 449-701, South Korea
{Maria, Saad, Sylee, i30000}@oslab.khu.ac.kr, Yklee@.khu.ac.kr

Abstract

Large-scale distributed systems, such as ubiquitous computing environments, require a service delivery mechanism in order to keep track of the vast set of services offered and make them available to interested clients. The amount of services and clients, their context, and loose coupling between them makes service delivery in ubiquitous environments different from other systems. This paper presents a solution to overcome these issues by utilizing the underlying ontology and semantics for service lookup. Access control over context data is also considered by specifying dynamic policies at the system and service level.[1]

1. Introduction

Ubiquitous computing [1] aims at seamlessly integrating the physical and computational worlds to enhance the user experience and provide services attentive to user's preferences in a given context. A service delivery mechanism facilitates such interactions by relating clients to the desired services.

In a generic service provider / consumer environment, the services are published and discovered based on syntactic parameters i.e., the consumer can lookup the service solely basing on syntax queries. In a context aware system, a vast set of heterogeneous context services and clients are present and only syntax-based queries might not be able to locate the desired service even if it exists; the syntactic representations of the 'same concept' might differ at the clients and the services. Another related aspect is that the needs/requirements of clients and the availability of the services are a function of their respective contexts. Service matching decision should not only base on the attributes in the service request but also on the context in which the request was made, since the requirements may change with the changing context. Similarly, a service might be available only under a given set of context and remain unavailable otherwise.

A pervasive context aware system senses information and infers useful contextual situations in a user-centric manner; it can easily become invasive from pervasive if access to such information is not carefully monitored. There is a need to enforce access control over the services and information offered by the system. The presence of context delivery services at the user interface level makes them a logical choice for this task.

While context awareness puts additional constraints on service lookup and access, it contributes towards the solution to these problems as well. We propose a context-based service delivery and access control mechanism in section 2. Related work is discussed in section 3 and we conclude our discussion in section 4

2. An Architecture for Service Lookup and Access Control

Context delivery services provide discovery and lookup functionality to locate appropriate context providers and deliver them to the client applications in accordance with the access control policies. Context providers register with the context delivery/registration service to announce the functionalities and information they have to offer. It is assumed that underlying domain ontologies [2] exist; they associate semantic meanings with the attributes registered by a service and help in overcoming the problems which arise when the clients and services use different syntactic representations. Context information associated with the services and clients is used to identify the active services as well as the services that are relevant to clients' needs in

[1] This research is supported by the Ministry of Information and Communication, Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment) in collaboration with ICU

a given context. Context also plays an important role in identifying the access control policies and the client's credentials. The details are given in the following subsections.

2.1. Narrowing the Search Scope

In a context aware environment, a vast set of client's and services are present. It is imperative to limit the scope of search for a given client's query. In our approach, following parameters are considered for this purpose:

- *Services' Context* – identifies active services in the given context
- *Semantic Attributes* – helps identify similar concepts among the service attributes and query parameters
- *Client's Context* – narrows down the matched services basing on client's context

Under normal circumstances, the context delivery module needs to search for a match among all the available services. By considering the above mentioned parameters, the search is narrowed down to a set of services that are active in the given context, whose published attributes are a semantic match to service's query, and they are relevant to the client basing on the client's context. Figure 1 illustrates this concept.

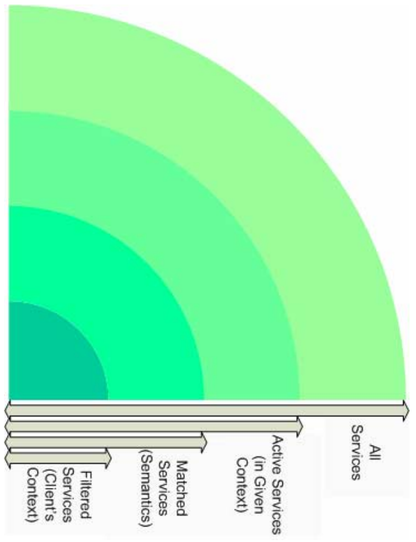
All Services
Active Services (in Given Context)
Matched Services (Semantics)
Filtered Services (Client's Context)

Figure 1. Limiting search scope from all services down to a set of filtered services

2.2. Dynamic Policy Selection

In a context aware environment, there is a need for dynamic privacy and access control mechanism as the security requirements vary with time. Since it is not feasible to define a single, system-wide policy that can cater for all kinds of changing situations at run-time, we define a pool of policies specifying various levels of security requirements. Each policy has an activation condition which becomes true under specific system context/ condition at runtime. Incorporating access control requires additional information inflow on behalf of the clients i.e., clients need to provide credentials to satisfy a given policy for service access. We can minimize this by considering client's context as implicit credentials in a service request (e.g. user location, identity, status). Clients are prompted to provide credentials explicitly only if the implicit credentials are insufficient to access the service. In case access control is not needed over some general purpose services (such as weather information, books in the library, goods available in the market etc), these services are readily available to all clients. The role of context in access control is:

- *System's Context* – helps in selecting the system policy applicable in the given situation from a policy pool
- *Client's Context* – Serves as the implicit credentials which are evaluated against the active policy to access a given service/resource

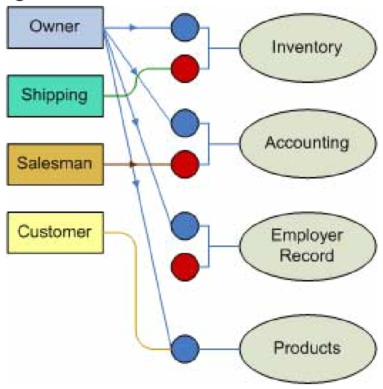

Figure 2. Relation between clients' credentials and service access restrictions

Figure 2 gives an example of various credentials and services available in a shopping area. Each service can register multiple interfaces differing in the number of functions offered or granularity of information available through them. Basing on the client's credentials, a service interface is selected from the available interfaces and returned to the client. By utilizing context-based policy selection and credential specification, a dynamic and interactive access control mechanism is put to effect.

2.3. Architecture

The architecture for service delivery in context aware environments should consider: semantic matching of services to overcome various syntactic representations at the client and services; context-based service filtering to narrow down the scope of search; dynamic policies to incorporate changing privacy requirements in a pervasive environment; minimum information inflow on behalf of the client to interact with the system in a secure manner. Figure 3 shows various components of the context delivery system that provide the aforementioned functionalities.

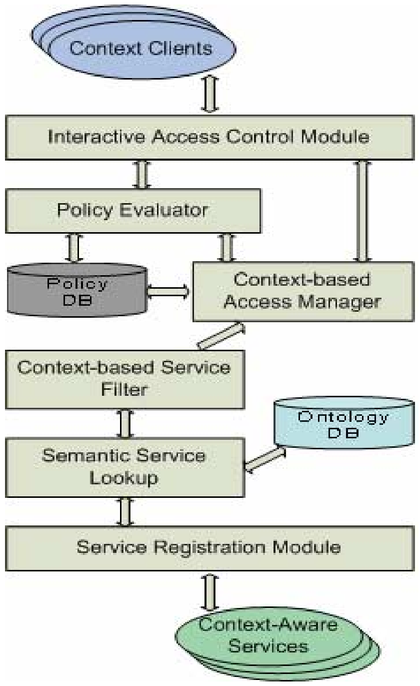
Context Clients
Interactive Access Control Module
Policy Evaluator
Policy DB
Context-based Access Manager
Context-based Service Filter
Ontology DB
Semantic Service Lookup
Service Registration Module
Context-Aware Services

Figure 3. Service delivery architecture

Context-aware services register their attributes and service proxy with the *Service Registration Module.* The attributes can be service name, type, location, contextual information provided by the service, semantic structure of the information etc. Other functional attributes can also be associated with the services to make the search more specific. It contains the basic functionality of a lookup service or service broker and serves as a registry for all the available context services in the system. It provides service registration, lookup, event management, and lease management.

The modules that contribute towards semantic search and context-based service filtering are *Semantic Service Lookup* module, *Context-based Service Filter* and *Ontology DB*. Ontology database contains the domain specific categorization of the contextual information. The *Semantic Service Lookup* module utilizes this database to trace services that provide the information client is looking for, even when the syntactic representation of the information at the client and service is different. For example, if a client requests a '*Temperature*' service, and the system contains a '*Weather*' service, the semantics will help identify that '*Temperature*' is a specialization of '*Weather*' information. Similarly the request can be matched to an '*Environment*' service since '*Temperature*' is an attribute of the environment. The search is performed only over the active services in the given context. Finally, *Context-based Service Filter* selects the services that are relevant to the client's context and filters out the rest. For instance, if a user is at '*Location-A*' and he wants to use a printer, the printer services at locations far from '*Location-A*' are filtered out in this process.

Once the services fulfilling the client's request and relevant to client in the given context are identified, the access control modules assess whether the request should be granted or not. The modules that take part in this process are *Context-based Access Manager*, *Policy Evaluator*, *Interactive Access Control Module* and *Policy DB*. Policy database contains the system level policies as well as optional service (context provider) level policies, defining the conditions to access a specific service provided by the system. *Context-based Access Manager* has two-fold purpose. It selects active system policy in a given context as well as computes the implicit client credentials basing on the client's context, as mentioned in the previous section. *Policy Evaluator* evaluates the credentials against the policies selected by the access manager. It implements various functions presented in [3]. Basing on the evaluation result, one of the service interfaces is returned to the client Incase the implicit credentials are not enough to fulfill client's request, *Interactive Access Control Module* requests the clients for missing credentials.

With the help of the proposed architecture, we can improve the service selection process by utilizing additional, but readily available, information such as semantics defined in the ontology and context information of the client and services. The policy selection and credential specification mechanism considers the dynamic nature of a context-aware environment, and contributes towards transparent and non-intrusive access control over services.

3. Related Work

The purpose of a service delivery mechanism is to enable coordination among the service providers and consumers. A number of middleware solutions use CORBA based coordination and communication [4] which provides dynamic resource discovery but the context delivery mechanism only deals with the syntactic comparison and does not utilize context information or semantics of attributes in the search. Web services and UDDI based approaches [5] provide a platform-independent mechanism for service description, discovery and usage but it limits the flexibility and expressiveness in the service descriptions to simple attributes: URL and string comparisons. Currently, efforts are underway to enable semantic matchmaking capabilities in the UDDI repository for web services [6]. In [7], the authors give architecture and implementation regarding ontology based service discovery. They provide the degree of matching such as exact, partial or no match. However, by incorporating access control features in the service delivery, the overall system integrity can be improved. Recently, focus is given to provide specific mechanisms for handling access control in context aware environments. In [8], access control is maintained basing on the context of the user. Anyone can have a specific access right by just being in a contextual situation which makes it more suitable for closed environments instead of open context aware system. In [9], the authors extend the existing access control model by incorporating roles and delegation mechanism but they only focus on controlling access to devices such as printers and do not define access control over information which is more critical in a pervasive environment [10]. Another aspect that differentiates our work is that in existing approaches for access control, the policies are mostly static which can not be modified at run-time to incorporate changing system privacy requirements.

4. Conclusion

In this paper, we have considered two main issues related to service delivery in context aware environments. Firstly, semantics defined in the ontologies are utilized to associate semantic meanings with the service attributes. This enables matching of client request to a specific service even when the syntactic representation at both is different. Context information is utilized to filter out unrelated services from the search. Secondly, access control over services is maintained with the help of a policy-based access control mechanism. The difference from existing approaches is the dynamic policy selection basing on system's context and implicit credential specification basing on client's context. This results in more specific service selection and a secure context aware environment. We have implemented our approach on top of the existing discovery and lookup service of Jini [11]. Currently, efforts are underway to find a suitable description language for policy specification.

References

[1] Weiser, M., "The Computer for the 21st Century", Scientific America, Sept. 1991, pp. 94-104; reprinted in IEEE Pervasive Computing, 2002, pp. 19-25.

[2] W3C Web Ontology Working Group: The Web Ontology language: OWL. http://www.w3.org/2001/sw/WebOnt/

[3] Koshutanski, H., Massacci, F., "Deduction, Abduction and Induction, the Reasoning Services for Access Control in Autonomic Communication", proceedings of the 1st IFIP TC6 WG6.6 International Workshop on Autonomic Communication (WAC 2004), Berlin, Germany. Springer, October 2004.

[4] Yau, S. S., et al., "Reconfigurable Context-Sensitive Middleware for Pervasive Computing", IEEE Pervasive Computing, joint special issue with IEEE Personal Communications, 1(3), July-September 2002, pp.33-40.

[5] Keidl, M., Kemper, A., "Towards Context Aware Adaptable Web Services", Proceedings of the 13th World Wide Web Conference (WWW), New York, USA, May 17-22, 2004, pp. 55-65.

[6] Pokraev, S., Koolwaaij, J., Wibbels, M, "Extending UDDI with context-aware features based on semantic service descriptions", International Conference on Web Services (ICWS), 2003

[7] Broens, T., Pokraev, S., Sinderen, M., Koolwaaij, J., Costa, P. D., "Context-aware, ontology-based, service discovery", EUSAI 2004, Springer, 2004, pp. 72–83.

[8] Corradi, A., Montanari, R., Tibaldi, D., "Context-based Access Control for Ubiquitous Service Provisioning", 28th Annual International Computer Software and Applications Conference (COMPSAC'04), September 28 - 30, 2004, pp 444-451.

[9] Kagal, T., Finin, L., Josh, A., "Trust-Based Security in Pervasive Computing Environments", IEEE Computer, Dec. 2001, pp. 154–157.

[10] Hengartner, U., Steenkiste, P., "Acccess Control to Information in Pervasive Computing Environments", 9th Workshop on Hot Topics in Operating Systems (HotOS IX), 2003.

[11] Sun Microsystems, Inc.: Jini^{TM} Architecture specification. http://www.sun.com/jini/specs/