DISCOVERING SEMANTIC WEB SERVICES IN FEDERATED DIRECTORIES

Michael Schumacher
University of Applied Sciences Western Switzerland, Institute of Business Information Systems, CH-3960 Sierre, Switzerland

Tim van Pelt, Ion Constantinescu, Alexandre de Oliveira e Sousa and Boi Faltings
Artificial Intelligence Lab, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Keywords: Semantic web services, distributed directories, intelligent agents, coordination, eHealth.

Abstract: This paper presents a flexible federated directory system called WSDir, which allows registration and discovery of semantic web services. Our directory system is used in a context where ubiquitous ehealth services should be flexibly coordinated and pervasively provided to the mobile user by intelligent agents in dynamically changing environments. The system has been modeled, designed and implemented as a backbone directory system to be searched by an infrastructure made up by such kind of agents coordinating web services. The system is modeled as a federation: directory services form its atomic units, and the federation emerges from the registration of directory services in other directory services. Directories are virtual clusters of service entries stored in one or more directory services. To create the topology, policies are defined on all possible operations to be called on directories. For instance, they allow for routed registration and selective access to directories.

1 INTRODUCTION

UDDI has become the de-facto standard to provide a general framework to describe and discover services and Web service providers. More specifically, WSDL descriptions of web services can be mapped to UDDI data structures, allowing to find web service descriptions using the standard UDDI query interface. Within the academic world, a number of approaches exist that try to build semantically enhanced discovery components on top of UDDI. (Kawamura et al., 2003) augments the standard UDDI registry APIs with semantic annotations. (Verma et al., 2004) uses a set of distributed UDDI registries as a storage layer, where each registry is mapped to a specific domain based on a registry ontology.

This paper presents a new federated directory (or registry) system called WSDir[1], which allows registration and discovery of OWL-S semantic web services (Martin et al., 2004). Our directory system is used in the CASCOM platform[2] where ehealth services should be flexibly coordinated and pervasively provided to the mobile user by intelligent agents in dynamically changing environments. We modeled, designed and implemented WSDir as a backbone directory system to be searched by an infrastructure made up by such kind of agents coordinating web services. This agent infrastructure therefore is deployed on mobile users: it queries our directory system for OWL-S service descriptions, composes them to achieve a target higher functionality and executes them. WSDir is currently being used in the trials of an ehealth emergency application.

We followed specific requirements for designing WSDir: i) it should have itself a Web service interface to be universally invoked; ii) it should be distributed; iii) the construction of the network should induce minimal overhead and should be scalable; also, the network should be robust to changes in topology and the number of interactions with the system; iv) the directory should allow a great number of services to be registered, and this in a very dynamic way, including lease times.

These requirements lead us to model WSDir as a federation: directory services form its atomic units, and the federation emerges from the registration of

[1]This work has been supported in part by the European Commission under the project grant FP6-IST-511632-CASCOM.

[2]http://www.ist-cascom.org

Schumacher M., van Pelt T., Constantinescu I., de Oliveira e Sousa A. and Faltings B. (2007).
DISCOVERING SEMANTIC WEB SERVICES IN FEDERATED DIRECTORIES.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 283-286
DOI: 10.5220/0002362902830286

directory services in other directory services. Directories are virtual clusters of service entries stored in one or more directory services. To create the topology, policies are defined on all possible operations to be called on directories. For instance, they allow for routed registration and selective access to directories.

The paper is organized as follows. In Sec. 2, we explain the service entries stored in WSDir. Sections 3 to 6 explain the concepts and architecture of WSDir. In Sec. 7, we give a concrete network architecture example based on WSDir. Finally, Sec. 8 concludes the paper.

2 SERVICE ENTRIES

We describe services using OWL-S (Martin et al., 2004). Internally, the directory system stores then service entries in the FIPA SL0 description language. The internal representation contains a subset of the information provided in the original service description, to be used to find matching services. In addition, the original OWL-S description is stored in a separate slot. This field is used to retrieve the original description, e.g., to retrieve the grounding(s) of a service at service execution.

In the following, we present the information that a service entry in WSDir contains:

ServiceCategories: entry in some ontology or taxonomy of services.

ServiceProfileURIs: set of profile URIs that point to an externally stored, but (web-)retrievable service profile.

ServiceProcessURI: process URI defined in the service description (can be empty).

ServiceGroundings: set of full-text service groundings (can be empty).

OWLSServiceDescription: original OWL-S service description as a full-text entry.

3 DIRECTORIES

A directory comprises a set of service entries which are managed by a collection of one or more directory services. All service entries, including directory service entries, are registered at a directory service as belonging to a specific directory. Such, directory services can form an arbitrary organisational structure (peer-to-peer, hierarchy etc.). Specifically, a directory can contain other directories, and a directory is supported by one or more directory services.

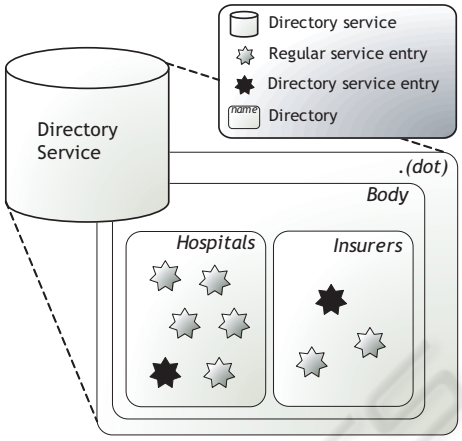

Figure 1: Visual recapitulation of directory system concepts.

The above is used to characterize directories by two different types of interactions:

- *Client-Directory* interactions: in which clients registering, deregistering, and querying the directory interact with directory services supporting the directory.
- *Director-Directory* interactions: in which directory services supporting the directories interact with one another to perform the internal management of the directory (e.g. federated queries).

4 DIRECTORY SERVICES

Directory services provide a Web service interface to a repository that holds service entries. The service entries in this store are all registered as belonging to a certain directory. The directory service forms the atomic unit of the directory federation. It allows clients to register, deregister, modify and search registrations in its repository. These registrations include service descriptions of services offered by clients as well as profiles of other directory service. By registering directory services in other directory service stores, the system becomes federated.

Figure 1 visually summarizes the relationship between service entries, directories and directory services. The directory service holds regular service entries and a directory service entry belonging to a Hospitals directory as well as entries belonging to an Insurers directory. Both directories are contained in the Body directory, which in turn is contained in the all-encompassing "." directory.

5 DIRECTORY OPERATIONS

The directory is able to handle five types of operations:

Registration: The *register* operation enables a client to register a service description entry into a *directory* for a time period given by a *lease-time*. The *directory-service* can return a *redirect* message pointing to another directory where the client could try to register its object.

Deregistration: The *deregister* operation deregisters a service that previously has been registered identified.

Modification: The *modify* operation allows modifying a registered service.

Search: The *search* operation looks in the directory for services that match a template. The request can possibly be forwarded to supporting directories, depending on the implemented search policy at the directory. As the internal service descriptions are expressed as SL0 expressions, the structured element used in the operation is also an SL0 expression. *Search-constraints* can be specified in order to restrain the search:

- a *max-time* specifies a deadline by which the constrained search should return the results.
- a *max-depth* specifies the maximum depth of propagation of the search to federated directories.
- a *max-results* element specifies the maximum number of results to be returned.

Get Profile The *get-profile* requests meta-information on a directory service such as its name and the policies governing the operations.

6 POLICIES

Directory services employ *directory policies* to regulate the operation of directories. Policies are defined per directory service in the *directory service profile* and determine the behaviour of a specific directory. Two types of policies can be distinguished:

- *Pro-active policies* are typically used for internal management of the directories.
- *Reactive policies* assign a behaviour to combinations of directories and operations; they are executed whenever a bound operation is called and defined as a triple: *(directory name, operation, policy)*.

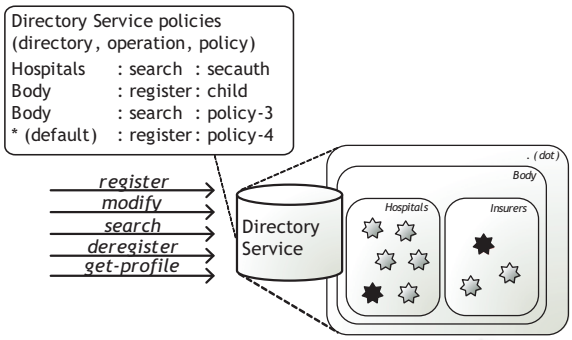

Figure 2: Example use of policies.

From the consequent application of policies, the network topology emerges. Policies can for example define how much entries can be registered per directory, which directories can be searched by which clients, and which types of services will be accepted. Each directory service can define its own policies or use one of the pre-defined policies. The only requirement on the part of a policy is that it can be executed. A straightforward example of a pre-defined policy is the *child/sibling* policy: this policy forwards all operations to both the known children as well as its known siblings.

Figure 2 shows an example of the reactive policies that are assigned to the various directories this directory service supports. In the illustration, when a client calls the search operation on the ”Hospitals” directory, a policy called ”secauth” will be applied. Such a policy could for example require the client to authenticate itself before it is allowed to query the directory.

7 AN EXAMPLE OF A NETWORK ARCHITECTURE

WSDir allows to setup flexible distributed directory systems, especially thanks to the mechanism of policies. We present here a specific application of WSDir to build a network of directory services which are modelled as a virtual tree with multiple roots.

The nodes of the tree are made up by the individual directory services. This hierarchical structure with multiple entry points effectuates:

- No replication or data is cached within the directory;
- Directory services will forward queries to other directory services;
- Query messages are checked for duplication;

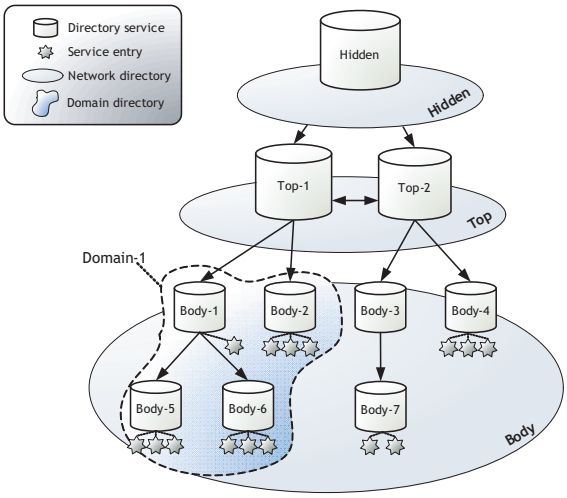

Figure 3: Network Topology.

- Identical results may be returned by one or more directory services for a single query.

Figure 3 illustrates our network topology. We distinguish two types of directories: network directories and domain directories. *Network directories* are a reserved set of directories that are used for the construction of the network. Among them, we can distinguish three different network directories:

- *Hidden* network directory forms the root of the federation by registering the top-level nodes of the network.
- *Top* network directory is visible to the network as being one of the roots of the federation multi-rooted tree.
- *Body* network directory is used for regular directory services that form the body of the multi-rooted tree.

Domain directories emerge from the registrations of service descriptions at the directory services that make up the directory system. By definition, domain directories are contained in the ”Body Members” directory.

At boot time, the directory makes use of a pre-defined network configuration to create a network topology. The configuration specifies management and data relations between members of the network. In a typical setting, the node at the highest level will be hidden to all nodes not belonging to the ”Top Members” directory.

To construct the network topology, WSDir employs a set of pre-defined pro-active policies. For instance, a specific policy would make search requests directed at a directory service be forwarded to its registered children and its known siblings.

8 CONCLUSION

WSDir has been tested thoroughly in a real distributed setting spread over different countries. The system has proven to be scalable and very stable. It is being used as backbone in a use case scenario in health emergency management. Future work could especially enhance security.

We currently employ standard security mechanisms for accessing the directory services. In particular, if a directory service requires protecting messaging from overhearing or if it would require privacy sensible data as parameters, the access to this web service will be based on HTTPS. In cases where no HTTPS is available, e.g., in most MIDP 2.0 implementations, we could couple WSDir with Guarantor agents (Bianchi et al., 2005) spread in the architecture in order to provide a secure tunnelling between secure ACL messages and HTTPS.

Another improvement could define security measures directly within the directory system by defining specific policies. A policy can be employed to restrict the right to perform a certain operation on a directory to only those clients that can provide the right credentials.

REFERENCES

Bianchi, R., Fontana, A., and Bergenti, F. (2005). A real-world approach to secure and trusted negotiation in mass. In *AAMAS ’05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1163–1164, New York, NY, USA. ACM Press.

Kawamura, T., Blasio, J.-A. D., Hasegawa, T., Paolucci, M., and Sycara, K. P. (2003). Preliminary report of public experiment of semantic service matchmaker with uddi business registry. In *ICSOC*, pages 208–224.

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., and Sycara, K. (2004). Bringing semantics to web services: The owl-s approach. In *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*.

Verma, K., Sivashanmugam, K., Sheth, A., Patil, A., Oundhakar, S., and Miller, J. (2004). METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *Journal of Information Technology and Management*.