An Architecture for Resource Discovery in Distributed Communications Systems

Temitope O. Olukemi

Department of Electronic and Electrical Engineering
University College London

Abstract: *As networks continue to evolve and become more distributed there is an increasing demand for a more rapid deployment of new services and feature into the networks. This presents an increasing challenge to network operators, as the speed of service deployment is a critical factor in today's competitive market. This paper presents an architecture that can be used in facilitating the fast introduction of new services into a distributed network environment. It describes how the concept of metadata trading can be used to provide rapid resource discovery within a distributed framework. The metadata is described using the extensible markup language (XML) which provides the framework with an open and extensible platform.*

1. Introduction

As the cost of network technology has been decreasing there has been a corresponding increase in the deployment of network infrastructure by network operators. At the same time there has been an increasing demand by network users for a greater variety of not only better services but also services which are more flexible and easily configured by the user. This has challenged operators to develop and deploy new services and new features into their network rapidly. This is done not only to satisfy their customers, but also to maintain some form of differentiation between them and their competitors in the increasingly competitive environment.

This vast distributed infrastructure and the need for rapid service development and deployment has greatly added to the management overhead of network operators. Many network operators now run large complex management platforms. The overhead of running these platforms is beginning to outweigh the overhead of provisioning the networks themselves. There is now an increasing need to define new management systems that are simple but yet flexible enough to allow rapid deployment of new features and services.

The notion of a Trader is to provide a framework whereby resources needed for these new services can be discovered at runtime. Trading decouples the clients and server objects by providing an intelligent directory service, the basis for the service discovery framework [1]. This intelligent directory service allows for the type of dynamic binding that is needed between clients and servers at runtime. The concept of the intelligent directory service is to provide a binding between clients and server objects based on the properties of the server objects rather than on their name. Thus it is possible to conceive of a particular service being implemented in a multiplicity of ways.

2. Metadata

The trader trades in metadata. Metadata is data about data. We use the term metadata in this instance because the data describing the resources in the network is self-descriptive. In present day distributed systems such as those based on CORBA (Common Object Request Broker Architecture), the capabilities of resources are described using the interface definition language (IDL). IDL describes the interface, specifying methods it supports, input and output parameters and exceptions that are thrown. For the framework being described here, IDL is not sufficient as it fails to provide any idea of the behavioural characteristics of the component. While it gives us a good syntactic definition of the interface it gives no information on the non-functional properties of the component such as management, dependencies, associations, security and performance [2]. To overcome this limitation of IDL, XML [3] is used to describe the metadata of the resources. XML provides a semantically rich as well as a syntactic structure that can be used for a complete description (capabilities and behaviour) of a resource within a network. XML also has other advantages:

- it is an open standard,
- a wide variety of XML tools already exist in various programming languages and platforms,
- it is extensible,
- through the use of document type definition (DTDs) or XML-Schema [4], it provides a means whereby the metadata can be validated.

3. The Architecture

The functional architecture of the resource discovery framework consists of three basic components namely: the trader, resources and clients.

3.1 Trader

The trader acts as a mediation service between resources in a network and clients that require those resources. The trader is built around the XSet database [5], Apache's Xerces-J parser and Apache's Xalan-J XSL processor. The trader is implemented as a distributed service in Java using remote method invocation (RMI). This makes it possible to have a federation of traders. Figure 1 shows the architecture of the trader.

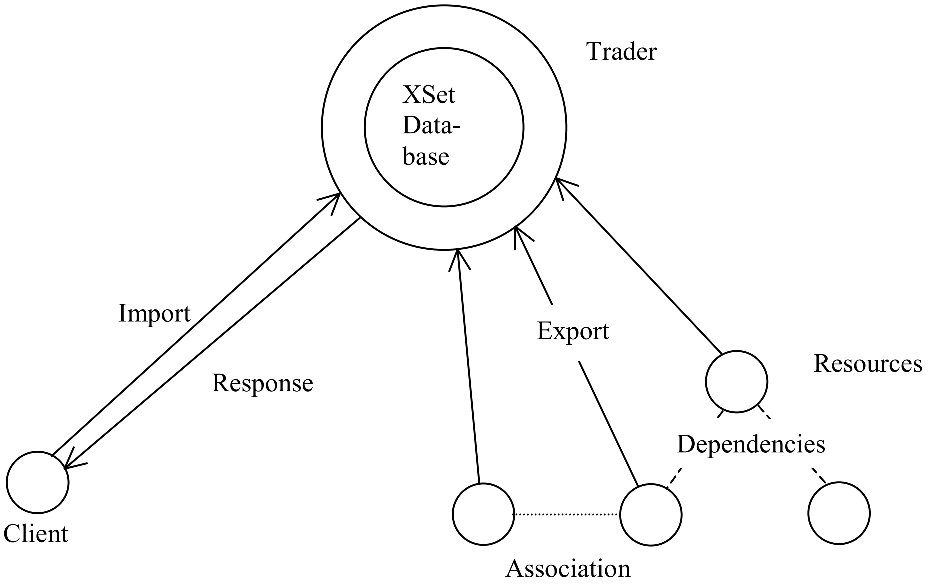

Figure 1 The Architecture of the Metadata Trader

There are two scenarios under which the trader provides services:

1. In the basic scenario resources within the network periodically advertise their metadata to the trader as XML documents. The trader stores the metadata in the XSet database. Clients can then query the trader for resources meeting certain set criteria. The trader returns to the client, XML documents of the metadata of the resource(s) that match the client's criteria.

2. A more advanced scenario is built upon the basic scenario. In certain contexts the client may require more information than just the metadata of a resource that matches its specified criteria. For example, consider an instance where a client wants to obtain a binding on an asynchronous transfer mode (ATM) virtual channel. The client will not only require metadata describing the virtual channel; it will also need metadata describing the virtual path that the virtual channel is a constituent of. In these instances the trader must not only to return the resources that match the criteria specified by the client, but it must also return other resources that are related to the requested resource (such as dependencies and associations).

Apart from mediation function the trader also provides a security function allowing clients to access services based on access rights specified in the clients import request. Presently the functionality of this is limited, as the trader presently does not have a means of determining if the client object is actually who it claims to be.

3.2 Resources

The resources are the various elements that make up the network. Resources can be viewed as service providers. These include routers, switches, processors, disk space, bandwidth, etc. Each resource within the network has associated with it a set of metadata. Figure 2 shows an extract of a DTD describing a resource.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD for CORBA Component Descriptor. The root element is
<corbacomponent>. Elements are listed alphabetically.
-->
<!ELEMENT client   EMPTY  >
<!ELEMENT componentfeatures
(inheritscomponent?,supportsinterface*,ports,extension*) >
<!ATTLIST componentfeatures
               name  CDATA    #REQUIRED
               repid  ID    #REQUIRED  >
<!ELEMENT componentkind  (service | session | process | entity
| unclassified) >
<!ELEMENT configurationcomplete   EMPTY  >
<!ATTLIST configurationcomplete
```

Figure 2 Extract of a Resource DTD

3.3 Clients

The clients as proposed in this architecture are service requesters. Using the trader the clients obtain bindings on various services provided by the resources within the system. Client import requests are sent to the trader. The information that is sent in the import request includes the access rights of the client and the query for the required resource. The query model used is an extended version of that provided by the XSet database [5]. The XSet query model supports tag matching and range queries. The model has been enhanced to also support dependency queries. Clients could be resources for other clients.

4. Conclusion

This paper has attempted to present a metadata trader, an architecture for resource discovery in distributed communications systems. As networks continue to grow and become more distributed, such architectures will become increasingly important if management overheads are to be kept to a minimum.

XML, being an open extensible standard, fits naturally into this trader architecture providing network operators with an open extensible platform upon which they can deploy new services and features rapidly into their networks. Presently the architecture only provides a resource discovery framework. There is no facility for secure communications across the RMI interface used. The further development and integration of the emerging XML standards such as XML-Signatures [6] will add robustness to the system. Specifically XML-Signatures will provide a means of determining the authenticity of a client requesting services from the trader.

References.

[1] V. Vasudevan "A Reference Model for Trader-Based Distributed Systems Architectures" Object Services and Consulting January 1998

[2] P. Mckee, I. Marshall "Behavioural Specification using XML" BT Advanced Communications Technology Centre 2000

[3] Extensible Markup Language, W3C recommendation February 1998, http://www.w3.org/TR/1998/REC-xml-19980210

[4] XML Schema Part 1: Structures W3C Working Draft April 2000 http://www.w3.org/TR/xmlschema-1

[5] A.D. Joseph, B.Y. Zhao "XSet: A Lightweight Database for Internet Applications" http://www.cs.berkeley.edu/~ravenben/xset/html/xset-saint.pdf

[6] XML-Signature Syntax and Processing W3C Working Draft July 2000 http://www.w3.org/TR/xmldsig-core