7 IN SEARCH OF THE RIGHT PARTNER

Simon Field and Yigal Hoffner
IBM Research, Zurich Research Laboratory
8803 Rüschlikon, SWITZERLAND
{sif,yho}@zurich.ibm.com

Finding matching customers and providers is an essential part of any dynamic e-business or virtual enterprise solution. Existing standards such as the CORBA Trading Service or emerging standards such as UDDI fall short of what is needed if the promise of dynamic e-business is to be fulfilled. This paper explores what is needed, and presents an implementation that satisfies those requirements.

1. INTRODUCTION: IN SEARCH OF THE RIGHT PARTNER

Web services are components with a description of their functional aspects and access protocols in Web Services Description Language (WSDL) that can be invoked over the Internet. Any attempts to make web services technically and commercially viable require powerful and sophisticated ways to describe such services and to advertise and search for them.

The **Universal Description, Discovery and Integration (UDDI)** standard (Graham 2001a,b; Tidwell 2001; UDDI 2001) provides the basis for an online directory for finding such services but falls short of what we consider sufficient for such an environment for a number of reasons.

Web services are likely to be complex entities with appropriately complex descriptions. Interface definitions in WSDL together with what UDDI calls "White" and "Yellow Page" descriptions are not going to provide a sufficient basis for searching for such services.

If the descriptions of the services are not well structured, textual searches will be needed to find matching consumer-provider pairs. Such searches are likely to require human intervention, and lack of standardised descriptions will adversely affect the efficiency of searching. Support for dynamic service composition demands that human intervention should be avoided, wherever possible, in the process of searching for web-services and configuring the consumer and provider resources for the services' consumption. More structured ways of describing web-services are therefore needed in order that matchmaking process can be automated.

Furthermore, an asymmetric search in which one side describes itself and the other describes what is it looking for, does not cater adequately for many real-world business scenarios where, in addition to the customer stating requirements of the

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35585-6_68

L. M. Camarinha-Matos (ed.), *Collaborative Business Ecosystems and Virtual Enterprises*

service, the service provider is likely to have requirements of the customer. The current UDDI model only supports an asymmetric approach to searching.

Some companies may consider aspects of their service description to be confidential, and might not wish to advertise full details of their services in advance, preferring to add further details as part of an exchange of information with potential customers. There is no provision in UDDI for such a staged revelation of information.

This paper provides a short overview of UDDI. A critique of the UDDI approach and a description of its shortcomings follow this. Finally, a description of a matchmaking engine that provides the facilities that UDDI lacks is given.

2. UDDI - MATCHMAKING FOR WEB SERVICES

The **Universal Description, Discovery and Integration (UDDI)** specifications define a way to publish and discover information about Web services. A UDDI registry can be seen as a directory service where comprehensive specifications of services rather than just names or addresses can be found and returned as a result of a query on the registry.

UDDI provides three basic functions (Fig. 1):

- **Publish**: How the provider of a Web service registers itself.
- **Find**: How an application finds a particular service.
- **Bind**: How an application connects to, and interacts with a Web service after it has been found.

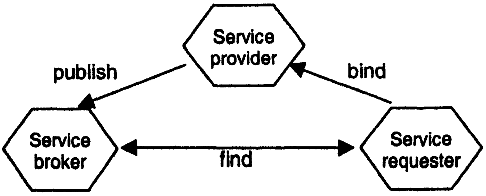

Figure 1 - The UDDI Service-Oriented Architecture (SOA) model.

Conceptually, a UDDI service contains three types of information, described in terms of telephone directories:

- **White pages**: Information such as the name, address, telephone number, and other contact information of a given business.
- **Yellow pages**: Information that categorizes businesses. This is based on existing (non-electronic) standards.
- **Green pages**: Technical information about the Web services provided by a given business.

More specifically, UDDI registries use four different levels of descriptions of services:

1. **Business information**: categories, contact date, URLs, etc. This can be regarded as the "White page"-level description.
2. **Service level description**. This can be regarded as the "Yellow page"-level description.

3. **Binding information**: how to bind the resources to enable communication between the service and its consumer. This can be regarded as the "Green page"-level description.
4. **Information about specifications for services**: This is metadata about the various specifications implemented by a given Web service. These are called tModels in the UDDI specification. This also can be regarded as part of the "Green page"-level description.

A matching reply from UDDI provides the business users with any or all of the above four levels of description.

3. A CRITIQUE OF THE UDDI APPROACH

UDDI is basically a standard for a directory providing limited search capability for services, having described and registered them using the prescribed levels of information.

In the commercial context of trying to sell services over the web, a number of things appear to be missing in UDDI:

- **Symmetry of selection**: UDDI allows the service provider to describe the service and the consumer to specify what they require of the service. This is not sufficient for finding business partners where complex services are involved. The selection process must be two sided where both sides have to describe themselves in terms of what they offer and may specify what they demand of the other party.
- **Power of description and power of requirement expression**: powerful languages are needed to describe complex services and similarly, powerful languages are needed to describe complex compatibility criteria, i.e. requirements on the description of the other party.
- **Ability to specify configurable services**: enabling the service description to be completed dynamically at the time of searching. This is needed for two main reasons:
  - It facilitates an up-to-date description of the service where the cost, availability or quality of service may be subject to variations. Such variations can for example be due to load, maintenance, etc.
  - It provides a way to specify services without having to enumerate all the options associated with them in the directory as this may overload the directory.
  - It provides a way to configure the service and the service consumer according to the needs and properties of the both parties. This facilitates personalisation of the service.
- **Facilitate the structured intervention of third parties**: the ability to ask for third-party evaluation of a service or a customer or the integral participation of third parties where combined/bundled services are concerned, for example,
- **Ability to distribute the matchmaking process**: allowing more commercially sensitive information to remain within the service-providing organisations, rather than requiring all descriptions to be shared centrally.

- **Ability to stage the matchmaking process**: allowing both parties to stage the revelation of information to each other, facilitating the sensitive handling of personal or commercially sensitive information, while also providing a more natural, phased approach to the search process through what is likely to be a complex information space.

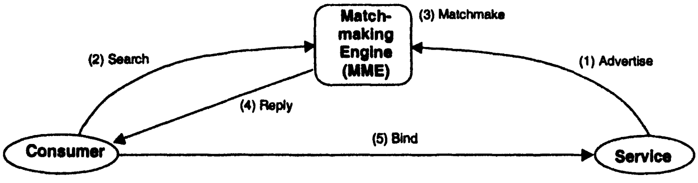

Figure 2 - The matchmaking service

4. THE IBM DYNAMIC MATCHMAKING ENGINE

A **Matchmaking Engine** (MME) is where advertisements are directed to and stored on a "permanent" basis, and where search queries are sent. The MME searches for an existing offer or offers that match each received search query.

A number of existing MMEs are used to support the electronic advertising and searching of goods and services (Bakos, 1991). Examples are IBM (Hoffner et al., 1999), Ariba (http://www.ariba.com), HP e-speak or MIT's Kasbah prototype (Chavez and Maes 1996). An elaborated categorisation of electronic marketplace systems is provided in (Schmidt, 1997).

The MME described here is an extension of the ANSAware/CORBA trading service (ODP 1995; OMG 1996). It is part of the IBM set of tools for the creation, management and maintenance of virtual markets and matchmaking spaces (Hoffner et al., 2000).

The following sections introduce the IBM Dynamic MME and show how it addresses the shortcomings of UDDI as stated in Section 1.

4.1. Matchmaking

In a matchmaking service, the service provider (or provider in this document) describes what is offered by the service (Fig. 2). The consumer initiates a search on the MME and the successful results of the matchmaking process are sent to the consumer. The consumer can then decide on the provider and bind to it.

Properties can be specified as fixed at advertisement time (static properties), or dynamic, in which case values will be provided at matchmaking time (dynamic properties). In the case of dynamic properties, the source of the property has to provide the information necessary for updating the property at matchmaking time.

4.2. Symmetric Matchmaking

The requirement stated in Section 3 of allowing both consumers and providers to select each other required a change in the basic functionality of the CORBA Trading Service by making the matchmaking process symmetric (Hoffner 1998). The Dynamic MME supports a symmetric matchmaking process as shown in Fig. 3. In the symmetric matchmaking process, both customers and providers describe what they offer and require of each other. In this case, the dynamic property update can include information from the prospective client, to enable the new value to take into account the specific details of the potential client.

The change to symmetric matchmaking necessitates the modification of the property-constraint language. Whereas in the CORBA trader, properties were by default service-provider properties, in the symmetric case it is necessary to distinguish between the properties of the consumer and the provider. In the symmetric MME each property is therefore preceded by an indication of its origin:

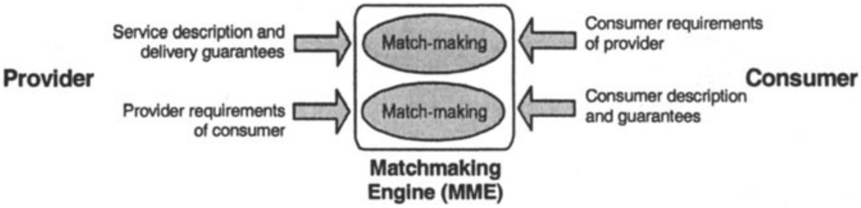
Service description and delivery guarantees
Provider
Provider requirements of consumer
Match-making
Match-making
Matchmaking Engine (MME)
Consumer requirements of provider
Consumer
Consumer description and guarantees

Figure 3 – Symmetric matchmaking process in the IBM Matchmaking Engine.

- Provider side properties are prefixed with "`provider.`"
- Consumer side properties are prefixed with "`consumer.`"

For example: `provider.ItemPrice` or `consumer.Number_of_Items`

4.3. IBM MME Property-Constraint Language

The matchmaking language has to be sufficiently powerful to express the type of properties that organisations are looking for when selecting a partner. The descriptions created with this language are capable of expressing complex attributes such as delivery dates with price tags attached and associated QoS parameters, etc.

Similarly to CORBA Trading Service, IBM MME Property-Constraint Language supports five built-in types: integer, float, string, chars and boolean.

In addition, IBM's MME language has a number of special data types and structures:

- **Date**: <[yy[yy][/MM[/dd]]]{sp}[HH[:mm[:ss[.SSS]]]]>. The *date* data structure has a set of operations allowing its components to be retrieved and modified. Addition and subtraction operations on dates are useful when constructing a new date from a given date and an interval. It is also possible to calculate an interval between two dates: *date1 – date2.*
- **Records:** It was decided to extend the language by introducing Records as a means of creating compound structures. Sequences of records and even nested sequences are possible with the IBM dynamic MME. Records provide an easy way to aggregate a number of attributes that belong to a single entity. This together with the notion of an array or sequence is particularly useful when addressing a group of people or services, or any entities with multiple attributes.
- **Indexable sequences:** Where the number of similar items which have to be dealt with by the matchmaking engine is not known in advance, it is necessary to introduce indexable sequences. Sequences can be constructed from all basic

types as well as from records. Elements of sequences can be referred to using their index in the sequence. This together with a FOR loop construct will enable a variable number of items to be dealt with.

The **IBM MME Script Language** provides the building blocks and composition rules for creating scripts for describing requirements, replacing the simpler constraint expressions of the CORBA Trading Service. The scripting language can be regarded as a simplified programming language consisting of:

- Data structures (similar to the ones used by properties), temporary variables and value assignments.
- Logical expressions (returning a Boolean value): e.g. *AND, OR, NOT,* >, <, <=, ==, *string* and *item containment.*
- Arithmetic expressions: e.g. *, /, +, -
- Conditional expressions: e.g. *IF-THEN-ELSE.*
- Control flow expressions: e.g. *FOR loops.*
- Built in functions: relating to the *Date type, absolute value, rounding, etc.*
- Comments.

The scripts created with this language are capable of expressing complex requirements and express trade-off between QoS, price, controllability etc.

4.4. Service Configuration Capability

Dynamic properties are properties whose value is not specified at advertising time but instead, the advertisement provides the information necessary for updating the property at matchmaking time. In the bi-directional matchmaking process, both clients and providers describe what they offer and require of each other. In this case, the dynamic property calculation can receive information from the prospective client, to enable the new value to take into account the specific details of the potential client.

When the MME comes across a service property marked as dynamic, it sends a request for a property update to the destination specified with the dynamic property (Fig. 4). If requested by the definition of the dynamic property, the MME will provide the required input properties as parameters to the dynamic property update-request. The value of the dynamic property will then be calculated and the newly derived property value is returned to the MME.

This allows the following to be done:

- It provides a way to specify services without having to enumerate all the options associated with them in the directory as this may overload the directory.
- It provides a way to configure the service and the service consumer according to the needs and properties of the other party. This facilitates dynamic personalisation of the service.

4.5. Multiple Party Matchmaking

The Matchmaking Engine can be further extended from two-party matchmaking to multi-party matchmaking, with each party supplying their own requirements and descriptions. The aim of multiple party matchmaking process is to facilitate:

- The incorporation of third party information providers such as rating agencies.
- Participation of third party where a combined/bundled service is concerned.

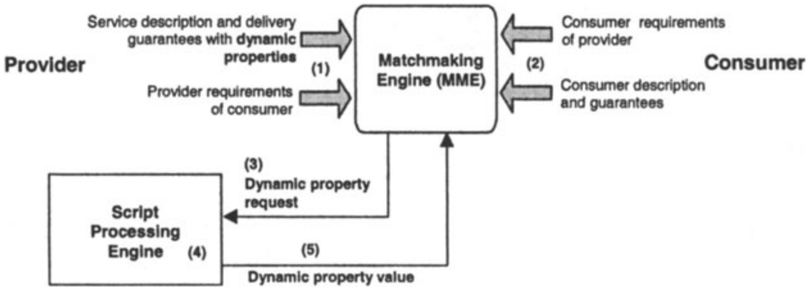
Service description and delivery guarantees with dynamic properties
Provider requirements of consumer
Provider
(1)
Matchmaking Engine (MME)
(2)
Consumer requirements of provider
Consumer description and guarantees
Consumer
(3)
Dynamic property request
Script Processing Engine (4)
(5)
Dynamic property value

Figure 4 – Dynamic service configuration via dynamic properties

4.6. Consumer-Service Provider Dialogue: Extensive Information Exchange

By creating a network of MMEs and distributing the information about the services, it is possible to direct different parts of an extended interaction to different MMEs, thereby creating a dialogue between the consumer and the network of MMEs. Each interaction with an MME can be seen as a phase in a multi-phase dialogue (Fig. 5).

This results in the distribution of the matchmaking process in both space and time. The detail of the service description, and consumer description required in phase 1 will be less than that required in subsequent phases. Physical distribution of the phases across a network of MMEs opens up the possibility that different organisations might be responsible for hosting different phases of the dialogue, and therefore different levels of information detail.

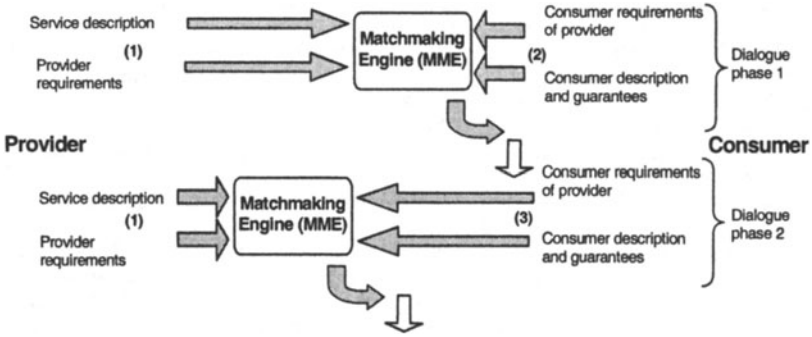

Figure 5 – Creating a phased dialogue with a network of MMEs

This might be of particular interest where commercially or personally sensitive information is involved in the matchmaking process. By dealing with more sensitive information in later phases of the dialogue, the matchmaking process can ensure that such information is only revealed to parties with whom earlier phases of the matchmaking have resulted in a successful match. In situations where a human consumer interacts with the MMEs, the results of each phase can be presented to the consumer, and selection of matching providers can be offered prior to proceeding to the next (more sensitive) phase.

Only those MMEs dealing with later phases of the dialogue will host the more detailed service descriptions. A distributed network of MMEs allows provider organisations to ensure that their most sensitive information remains in MMEs hosted by themselves, and not advertised in centralised MMEs, which may be shared with competitors in a virtual marketplace.

5. CONCLUSION

This paper has shown how a matchmaking engine, building on the ideas enshrined in the CORBA Trading Service, could play an important role in enabling Web Services to form the basis for dynamic e-business environments and virtual enterprises. We have identified some shortcomings in the UDDI approach to describing and finding service partners, and proposed some essential characteristics of a matchmaking service that would address these limitations. Finally, we have described the dynamic matchmaking engine, which has been implemented in Java and deployed in a number of customer projects. We believe that Web Services enhanced with these matchmaking capabilities represent a powerful foundation for the creation of dynamic e-business environments and virtual enterprises.

6. REFERENCES

Bakos Y. A strategic analysis of electronic marketplaces, MIT Quarterly, September 1991.

Chavez A, Maes P. "Kasbah: An Agent Marketplace for Buying and Selling Goods". In Proc. First Int'l Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, April 1996.

Graham S. The role of private UDDI nodes in web services, part 1: six species of UDDI. Web Services Architect, IBM Emerging Internet Technologies, May 2001a.

Graham S. The role of private UDDI nodes, part 2: private nodes and operator nodes. Web Services Architect, IBM Emerging Internet Technologies, May 2001b.

Hoffner Y, Schade A. "Co-operation, Contracts, Contractual Matchmaking and Binding". In Proc. 2nd Int'l Enterprise Distributed Object Workshop "EDOC '98," pp. 75-86. Piscataway: IEEE, 1998

Hoffner Y, Facciorusso C, Field S, Schade A. Distribution issues in the design and implementation of a virtual market place. Computer Networks 2000; 32:717-30.

ODP – Open Distributed Processing Reference Model, ISO/IEC 10476, ITU-T Recommendation X.900. 1995. Parts 1–3.

OMG – Object Management Group and X/Open. RFP5 Submission: CORBA Trading Object Service. Document orbos/96-05–6. May 10, 1996.

Schmidt B. Requirements for electronic markets architecture. Electronic Commerce 1997; 7:3-6.

Tidwell D. UDDI4J: Matchmaking for Web services, Interacting with a UDDI server, 2001. http://www-106.ibm.com/developerworks/webservices/library/ws-uddi4j.html. IBM, January 2001.

UDDI Site. The Universal Description, Discovery and Integration (UDDI) specification. 2001. http://www.uddi.org/.