Efficient and Transparent Web-Services Selection

Nicolas Gibelin and Mesaac Makpangou

INRIA, Regal Project, B.P. 105, 78153 Le Chesnay Cedex, France
firstname.lastname@inria.fr

Abstract. Web services technology standards enable description, publication, discovery of and binding to services distributed towards the Internet. However, current standards do not address the service selection issue : how did a consumer select the service that matches its functional (e.g. operations' semantics) and non-functional (e.g. price, reputation, response time) properties ? Most projects advocate automatic selection mechanism, advising adaptation or modification of the web-services model and its entities (UDDI, WSDL, Client, Provider). These proposals also do not take advantage of distributed-systems' state of the art, mainly with respect to the collection and the dissemination of services' QoS. This paper presents an extension of the initial model that permits automatic service selection, late binding and collection of metrics that characterize the quality of service. The extension consists on a web-service access infrastructure, made of web service proxies and a peer to peer network of QoS metrics repository (the proposal does not impose modification on UDDI registries or services). The proxies interact with common UDDI registries to find suitable services for selection and to publish descriptions. They collect QoS metrics and store them on a p2p network.

1 Introduction

The World Wide Web (WWW) has been used to store, exchange and provide static data. Over the time, new emerging technologies appeared. A broader variety of resources are increasingly being made available as Web services. For instance, in E-commerce applications, the WWW enables applications to applications (a2a) connection through multiple devices without being concerned with framework or languages heterogeneity. The basis of Web services like XML and WSDL for the service's description, UDDI and SOAP for services registry, discovery and communication, contribute toward making Web services a workable and broadly adopted technology.

The basic model specifies how to describe services and its interfaces, publish and discover methods. This initial model also provides abstractions to support multiple programming languages and run-time environments. However this model presents some drawbacks. First, it doe not support automatic selection of service when more than one of them satisfy the consumer functional properties With the growing popularity of Web services, finding relevant services become an important issue. This decision is left to the consumer who will handle it

B. Benatallah, F. Casati, and P. Traverso (Eds.): ICSOC 2005, LNCS 3826, pp. 527–532, 2005.

manually. Secondly, there is no mean to capture and/or exploit non-functional properties, such as quality of service, to help select the service that best suit consumer preferences. Another important drawback is the lack of a coherency mechanism that could guarantee the accuracy of information maintained by a UDDI Registry. It is reported in [1] that 48% of the production of UDDI registry have links unusable. Though this report dated back to 2001, we believe that the problem remains.

A number of authors have already identified some of these drawbacks [2, 3, 4, 5, 6, 7]. The solutions that were proposed either introduce modifications of components of the initial model or do not address the global picture. Modifications need to be agreed and integrated by all participants, which is almost impossible to achieve. In the other hand, solving part of the problem, for instance expecting the providers to provide the QoS, is not enough. Another consideration to keep in mind is the overall performance and efficiency of the solution.

To leverage the drawbacks of the initial web service model while letting its basic components functioning unchanged, we propose to extend the initial model with a web service selection and binding infrastructure that take care of at least the following functions: (1) automatic detection of broken web services references; (2) automatic collection of quality of service metrics for consumers; (3) consumers support of functional and non-functional properties for selection.

This paper is structured as follow: Section 2 describes in detail the new web service model, while Section 3 draws some conclusions.

2 Extended Model

We propose to extend the initial web service model with a selection and binding infrastructure, while leaving the UDDI Registry unchanged. Figure 1 shows where new components are located and the way they cooperate with existing components and within one another.

The proposed binding and access mechanism is achieved thanks to the introduction of two new components : web service proxy and p2p network of QoS metrics repository. In this section, we only focus on proxy mechanism.

A web service proxy offers: (1) to services' providers an interface to publish their descriptions to the UDDI registry, (2) to services' consumers an interface

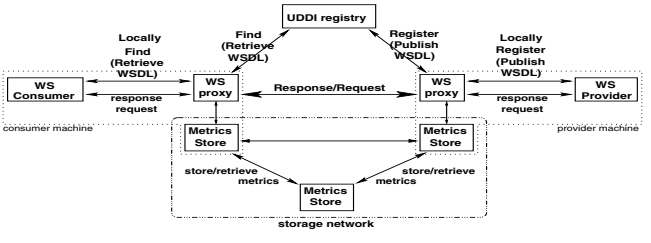

Fig. 1. Extended Web services model

to request the selection of services that best suit their functional properties and QoS requirements. With this new architecture, the requests of a consumer are forwarded to the selected service by the local proxy. The local proxy of the consumer cooperates with the local proxy of the provider to send request/response among the network. Overall, a group of cooperative web service proxies stand aside the consumers and providers and cooperate to relay (possibly modified) requests and responses to their final destinators which may be services or UDDI registries.

The set of cooperative web service proxies take advantage of their position (in between providers, registries, services and consumers) to observe ongoing activities and to collect information that can help evaluate various metrics characterizing the quality of the service offered to consumers. The collected metrics are stored in the p2p network of QoS metrics. The metrics' repository servers are in charge of the storage and the dissemination of the connected measures.

In the remaining of this section, we first specify what metrics we are considering to characterize the quality of service, as well as the means to collect these metrics. Then we discuss the main functions provided by the web service proxy. Finally, we present the selection algorithm that is implemented by web service proxies.

2.1 Quality of Service Metrics

In the extended model, we distinguish three categories of QoS metrics, depending mainly on the source of their measures. These are : service access metrics, feedback metrics, and service delivery metrics. The former class of metrics characterize the conditions for accessing the service. The metrics used by providers to characterize the conditions for accessing their services may vary from one provider to another; the measures for each service are supplied by its provider and may vary over the time. The feedback metrics measure the satisfaction or unsatisfaction of consumers. Finally, the service delivery metrics characterize the quality of the service offered by the underlying computation infrastructure. The metrics of the latter case can be computed automatically, while for metrics of the former two categories we need basics measures from end users.

Service Delivery Metrics. Unlike other extensions advocating automatic monitoring of Web Service QoS [3, 4, 8], we do not modify the provider or the consumer service. The monitoring is performed by web service proxies which can globally cooperate to obtain measurements for the following metrics: (1) service Load (average number of simultaneous; connections or requests to this service for some period of time) (2) response time (between consumer and provider); (3) service latency; (4) service throughput (average number of requests that the service can serve within a period of time); (5) service reliability (ability of a service to perform its required functions under stated conditions for a specified period of time. In our case, we monitor Mean Time between failure (MTBF) and Mean Time to Failure (MTF)).

All these metrics will be monitored and collected automatically thanks to the collaboration of web service proxies attached to providers and consumers. They

intercept all exchanges between the four entities that interact within the traditional web service models: providers, consumers, UDDI Registries and services.

Extracting Service Access Metrics. To permit a provider to supply the measures of the service access metrics defined for its service, we propose an extension of the classical ("portType, message, types, binding") common Web-Service Description Language (WSDL). For instance, the provider can specify service price, service resiliation penality, maximal delay before the service stop, compensation rate .... This article does not provide an exhaustive list of service access metrics that the provider can use to characterize its service. Figure 2 sketches the basic language constructions used to extend WSDL in order to specify provider-supplied service access metrics. This example defines two new metrics (standard to our model). The first, *servicePrice* describes the price a consumer must pay to access the service. In more complex cases, a provider can indicate the price on a per operation basis (described in the <wsdl:operation> section of WSDL). The second access metric of the example is *serviceDelay*; it measures the maximum activity time of the service. After that delay, the consumer must pay an other slice of time, or will be denied the use of the service. This example is not a full XML description of the proposed extension (no domain name is used), however it provides the basics keywords and language constructions a provider must use to describe its metrics.

```
<qos><metric name="servicePrice">
       <metricvalue name="price" value="200" type="int"/>
       <metricvalue name="currency" value="dollar-us" type="string"/>
     </metric>
     <metric name="serviceDelay">
       <metricvalue name="activitydelay" value="10" type="int"/>
       <metricvalue name="unitofmeasure" value="minutes" type="string"/>
</metric></qos>
```

Fig. 2. WSDL extension example

Consumer Preference for QoS. The web-service user can specify preferences to contribute to the selection mechanism. For instance, it will be able to specify metrics like "the least expensive service which exists", "the maximum price he accepts to pay for the service", or "the maximum response time he wants the service can satisfy". For that, the user sends an extended request specifying its preferences, to the proxy which store them. The proxy parses the request to extract consumer preferences and then clean the request to forward it to the traditional UDDI registry.

2.2 Web Service Proxy

The web service proxies are located on both provider and consumer side, providing classical retrieve/publish mechanism. The proxies are UDDI compliant, thus consumers and providers never directly contact the registries. In the sections below, we depict the extended retrieve and publish mechanism.

New Retrieve Procedure. In the first step, the proxy forwards the consumer request to a public UDDI register to find all the **serviceKey** matching the request and wait for the UDDI registry response (which may contain more than one service reference)

The second really important step checks if services are available. The proxy requests the UDDI registry to retrieve the localisation of all services, and then cooperate with monitoring network to determine wheather service remain available, eliminating no longer accessible or unavailable services.

Then, the proxy retrieves the available QoS of each service, using the **serviceKey** as hash key in the store network and starts the selection algorithm for each service to compute the QoS value of services. The algorithm returns a list of matching services in the order of best-matching first. The proxy then retrieves the service WSDL description.

Finally, the proxy sends a fake localisation to the service consumer (http://localhost:port/serviceKey/). The consumer can now communicates with the distant service, but all requests are forwarded by the proxy. As we will see, with this mechanism, we can provide transparent QoS measurements.

New Publish Procedure. From the provider side the publication procedure does not really change. When the proxy receives a publication request containing the service description WSDL, it first parses the description to extract provider supplied QoS information (Figure 2); then, it requests QoS monitoring network to store the extracted metrics. With this mechanism, we can follow the evolution of the price of the services such as to enable the selection of the cheapest service. Once the analysis is carried out and the QoS information propagated in the network, the proxy forwards the publication request and the WSDL description to the UDDI registry indicated by the provider.

2.3 Selection Mechanism

The selection algorithm is used to rank services. For each potential service, the protocol compute a rank value used to sort services. We can now describe the selection algorithm applied when a consumer sends a binding request to the proxy.This mechanism comprises the steps summarised below: (1) the consumer sends a request to the proxy with its preferences; (2)the proxy extracts the preferences and forwards the request to one or more UDDI registry indicated by the consumer; (3) the UDDI registry respond with a server list of potentials matching services (4) the proxy then retrieves QoS metrics of all the listed services, (5) finally, the proxy computes the algorithm and send back the response to the consumer.

3 Conclusion and Discussion

The paper proposed an extension of the web services model to enable automatic and transparent selection and binding to services that best suit the consumers

functional and non-functional requirements. This extension is built thanks to the web service proxy facility and a P2P network of QoS metric repository. Web services proxies associated with services providers and consumers offer standarized interfaces to permit them to interact with UDDI registries and Services, and take advantage of their position to collect measures for QoS metrics. Unlike most proposals that address the quality-based selection issue, the solution described in this paper does not require change on the UDDI registry or Services. It does not even require specific contribution from the services providers and consumers, but service access conditions for providers and annotations for consumers.

To access the benefits of the proposed selection and binding infrastructure, we plane to conduct a complete evaluation. Based on the results of some basic experimentation, we believe that this mechanism can improve the quality of the service experienced by the consumers. We also believe that this mechanism will help improve considerably the consistency of UDDI Registries as perceived by consumers.

References

1. Clark, M.: Uddi - the weather report. http://www.webservicesarchitect.com/content/articles/clark04.asp (2001)
2. Liu, Y., Ngu, A.H., Zeng, L.Z.: Qos computation and policing in dynamic web service selection. In: WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, NY, USA, ACM Press (2004) 66–73
3. Ran, S.: A model for web services discovery with qos. SIGecom Exch. **4** (2003) 1–10
4. Day, J., Deters, R.: Selecting the best web service. In: CASCON '04: Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, IBM Press (2004) 293–307
5. Mukhi, N.K., Plebani, P.: Supporting policy-driven behaviors in web services: experiences and issues. In: ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing, New York, NY, USA, ACM Press (2004) 322–328
6. Maximilien, E.M., Singh, M.P.: Reputation and endorsement for web services. SIGecom Exch. **3** (2002) 24–31
7. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, ACM Press (2003) 411–421
8. Maximilien, E.M., Singh, M.P.: Toward autonomic web services trust and selection. In: ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing, New York, NY, USA, ACM Press (2004) 212–221