



## **A Peer-to-Peer Service Interface for Manageability**

Vijay Machiraju, Akhil Sahai  
Software Technology Laboratory  
HP Laboratories Palo Alto  
HPL-2001-61  
March 23<sup>rd</sup> , 2001\*

E-mail: {vijaym, asahai}@hpl.hp.com

management,  
e-services,  
peer-to-peer,  
manageability,  
quality

# A PEER-TO-PEER SERVICE INTERFACE FOR MANAGEABILITY

Vijay Machiraju, Akhil Sahai  
E-Service Management Project  
E-Services Software Research Department  
HP Laboratories, 1501 Page Mill Road, Palo-Alto, CA 94034  
{vijaym, asahai}@hpl.hp.com

## 1. Introduction

Various interfaces and protocols are being defined for seamless composition and inter-operation of web services. Some of these are used by web services to discover each other (e.g., UDDI [1]), some of these are used by services to express their functionality to each other (e.g., WSDL [2], e-speak CDL [3]), and some of these are used to invoke operations on each other (e.g., SOAP [4]). While all of these are essential for easy composition, there are still many aspects that services need to agree upon in order to interoperate. One prominent example of such an aspect is manageability.

There are two interpretations of manageability for services. *Manageability from a management system's perspective* refers to whether a service provides sufficient information (events, measurements, and state) and control points (lifecycle control, configuration control, etc) to a management system so that it can be effectively monitored and controlled. There have been many efforts – CIM from DMTF [5], Manageability Service Broker from Open Group [6], and Java Management Extensions from SUN [7] to name a few – for standardizing the interface and protocol between managed services and management systems. So far, management systems and these standards have been focused on managing applications within an enterprise. As web services become more prevalent, cross-enterprise management of federated services will become increasingly important.

There is a second notion of manageability – *manageability from a peer service's perspective*. Figure 1 shows two peer-to-peer services interacting with each other. An interaction is any form of communication between two services. This could mean a single-step transaction (e.g., a login to a book-selling service), a sequence of related transactions (e.g., logging in, adding books to shopping cart, and checking out), or even business-level processes perceived by the client that may involve manual steps (e.g., the process from ordering books to final delivery of the books). For every step in an interaction, one of the two services initiates the request and the other executes the request. We refer to the service that initiates the request as the *consumer service* and the one that executes the request as the *provider service*. The role of a service could change over the course of an interaction.

From a consumer's perspective, a *provider service is manageable* if the latter offers sufficient visibility and control over itself and over the interactions it executes. For example, a provider that provides information about the progress of a consumer's ongoing interactions or an ability to escalate their speed is more manageable than a provider that does not. From a provider's perspective, a *consumer service is manageable* if it can offer enough information about its service usage back to the provider. For instance,

---

a consumer that can be queried about its perception or quality of experience is more manageable than a consumer that cannot be. Manageability interfaces capture the functionality that should be offered by providers and consumers to each other in order to be manageable.

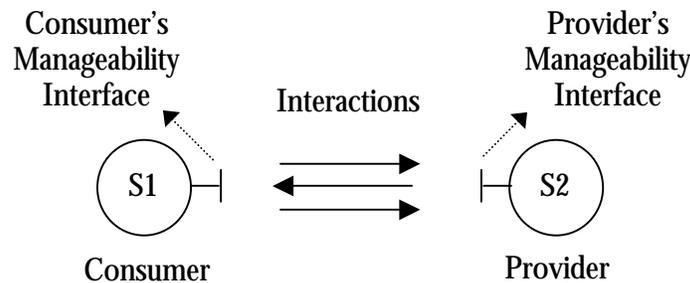


Figure 1: Peer-to-peer services, their roles, and manageability interfaces

There have been no efforts to standardize interfaces and protocols of this nature between web services. In this paper, we first motivate the need for such interfaces by explaining their benefits. We then list a set of elements that should be part of these interfaces. Finally, we conclude with some suggestions for how to approach the standardization.

## 2. Need for Peer-to-Peer Service Manageability

We envision a world where services will be capable of advertising, discovering, composing, and using each other to execute their functionality. Providing interfaces for manageability between services would help in realizing part of this vision.

In the discovery-phase or pre-composition phase, a consumer can use manageability interfaces of providers to obtain valuable information about their quality of service - performance, availability, reliability, and service-level guarantees. Brokers and other discovery services help clients select services. However, the selection criteria are currently limited to price and functionality. Standardizing manageability interfaces will facilitate negotiation and selection based on a whole new set of features. With this additional information, consumer services will be able to assess if they can guarantee end-to-end quality to their own clients taking into account the quality levels guaranteed by their provider services. When services are composed, it is not only the functionality that gets composed; the service-levels, performance, and availability are composed as well. Hence it is only natural for services to expose these details so that a consumer can make informed decisions before composition.

In the execution phase or post-composition phase, a consumer that uses a manageable provider has better visibility and control over the latter; and hence will be able to make better decisions. For example, a consumer that needs faster service can escalate its interactions to the next tier of service temporarily. Similarly, it can request long-lasting transactions to be shutdown so that they can be executed on a different service. Consumers will be able to compare different services for their quality and performance.

Third party rating services can be formed that just collect and sell ratings of various services for their quality. A need for third party mediators would also arise for mediating and monitoring compliance of service level agreements between other services. A real life analogous example would be the credit rating companies and the manner in which they settle disputes. However, the dynamics in case of web services would be different and so the third party mediators have to monitor the compliance and take corresponding actions in real time.

Being manageable is advantageous to the provider too. Manageability is a new differentiator for service providers. Higher levels of visibility and control can be provided to valued customers or at higher prices. We have seen various examples of ad hoc manageability offered by various services to stay competitive. For example, Fedex offers tracking of their shipping transactions (example of visibility). Most of the on-line e-commerce sites provide some form of a cancellation feature so that a service or product can be cancelled after being purchased. Web services have already realized the importance of accountability and guaranteeing service levels. A standardized manageability interface helps in projecting all these features of a web service.

One of the major drawbacks of current instrumentation and management technologies is the lack of an end-to-end solution. Once a service crosses a service provider (into a client or a supplier), the provider has no visibility or control on what happens with the interactions. Management systems internal to a service provider find it difficult to analyze if a failure or service-level violation is attributed to internal business logic or to a supplier. Having manageability interfaces on all services helps in solving this problem and in facilitating cross-enterprise management systems.

### 3. Elements of Manageability

Manageability is measured in terms of two factors – *visibility* or ability to observe and monitor, and *controllability* or ability to influence and change. One way to classify the functionality offered through a manageability interface is to differentiate elements of visibility from elements of control. Another dimension of classification is *service-level* versus *interaction-level*. Service-level interfaces are used to obtain information about or execute actions on the overall service. Interaction-level interfaces are used to obtain information about or control specific interactions between services.

As we have already explained, a service could act as a provider, or as a consumer, or both. The type of manageability depends on the role of the service. Table 1 shows examples of functionality that could be offered by a provider to be manageable. Table 2 shows some elements of consumer-side manageability interface. Detailed discussion on each of the items listed in these tables is beyond the scope of this position paper.

<b>Service-level Visibility</b>	<b>Interaction-level Visibility</b>
Nature of visibility and control supported.	Expected time an interaction will take to execute prior to submission.
Quality levels – performance, availability, and reliability – that can be guaranteed by the provider.	Expected time an interaction in progress will take to complete. Track an interaction in progress.
Quality levels currently being guaranteed to the consumer.	Amount of time a completed interaction took.
History of quality levels with the consumer.	History and statistics about past interactions.
History of quality levels with all the consumers.	Quality levels being guaranteed on certain types of interactions to the consumer.
Generic service metrics such as number of current consumers, average turn-around time etc.	
Service specific metrics – e.g., books sold per second for a book-selling service.	
<b>Service-level Control</b>	<b>Interaction-level Control</b>
Change current quality levels being guaranteed to the consumer. For e.g., change the consumer from bronze to gold customer.	Abort an interaction. Suspend an interaction. Resume a suspended interaction.

Report a consumer-perceived service-level violation to the provider. Ask for an explanation. May result in compensation according to the contract.	Change the desired quality level for an interaction.
--	--

**Table 1: Elements of a server-side manageability interface**

<b>Service-level Visibility</b>	<b>Service-level Control</b>
Nature of visibility and control supported.	Slow-down the request rate.
Quality levels perceived by the consumer.	Fail-over to a different service.
Quality levels currently being guaranteed to the consumer.	
<b>Interaction-level Visibility</b>	<b>Interaction-level Control</b>
Perceived quality levels for a particular interaction or a class of interactions.	Re-issue or restart an interaction. Suspend an interaction.

**Table 2: Elements of a consumer-side manageability interface**

## 4. Realizing the Vision

Realizing service manageability would require involvement and standardization on two fronts:

1. Define a set of standard terms, conversation definitions, and communication messages to support manageability of web services. Services first need to agree upon terms such as quality of service, contracts, and transactions. The next step would be to define conversations that should be supported by services to be certified manageable. For example, conversations for negotiating quality of service and for canceling transactions should be standardized. These conversations should cover all the types of visibility and control that can be offered by web services in a generic manner. Technologies such as XML, WSDL, and current management standards such as CIM would be helpful in defining these vocabularies and conversations.
2. Standardize extensions to current communication protocols such as SOAP to include manageability information. Services could communicate valuable information about interactions to each other by exploiting the messages that they exchange. For example a message from a service can include expected time of completion as part of its response. Headers that capture information such as quality level expected and conversation context can be standardized for every request. Similarly, quality level delivered, response time information, and conversation context can be part of every response header.

Coordinated efforts are also needed in the area of security for web services. A good foundation of security is a pre-requisite for services to be able to offer visibility and control over their functionality.

## References

- [1] Universal Description, Discovery, and Integration, <http://www.uddi.org>
- [2] Web Services Description Language, <http://www-106.ibm.com/developerworks/library/w-wsdl.html>
- [3] Conversation Definition Language, <http://www.espeak.hp.com>

[4] Simple Object Access Protocol, <http://www.w3.org/TR/SOAP/>

[5] Common Information Model, <http://www.dmtf.org/spec/cims.html>

[6] Manageability Services Broker, <http://www.opengroup.org/management/msb.htm>

[7] Java Management Extensions, <http://java.sun.com/products/JavaManagement/>