Distributed Systems Management for Enterprise Web Services Environment

Chia-Feng Lin
Department of computer science, National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300, ROC
teralin@gmail.com

Shyan-Ming Yuan
Department of computer science, National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300, ROC
smyuan@gmail.com

Ruey-Shyang Wu
Department of computer science, National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300, ROC
ruey@cis.nctu.edu.tw

Kuan-Yu Chen
Innovative Digitech-Enabled Applications and Services Institute, The Institute for Information Industry
11F, No. 106, Section 2, Heping East Road, Taipei, Taiwan, R.O.C
koumei@iii.org.tw

Abstract

In this research, we simplify Web Services Distributed Management (WSDM) interfaces developing effort. The interfaces are generated by program code that must be created during development. Developers do not have to understand all Web services standards. On the other hand, the management system should provide message flow oriented management atomically without modifying service code. Enterprise can control all flows and review them at any time. Enterprise can build a quickly, efficient and extensible management system in the Web services environment

Keywords: Web service, Agent, WSDM, Hook, API

I. Introduction

As Web services become pervasive and critical to business operations, the task of managing Web Services and the Web services architecture will be imperative to the success of such operations. Management in this case is defined as a set of capabilities for; discovering the existence, availability, health, and usage, as well as the control and configuration of resources, where resources are defined as Web services, components of the Web services architecture, and roles undertaken in this architecture.

System management is not easy because of the service composition dynamically and loose coupling. There are many uncertain events, exceptions and faults in this environment. Once error happened, the system administrator should identify where the problem is and find out the solution quickly. However, it usually takes a long of time to realize error happened and identify problems. To solve the problems quickly, an efficient management system is necessary. It can collect information from every application and device in the system. When error occurred, management system will detect it, get fault information and coordinate the related data for analysis. Therefore, system administrator can identify the problems and find out the solution to eliminate error quickly. Distributed system management is harder than single system management because more systems involved. It costs more time to find out problem because each system should be checked carefully when error occurred.

To manage huge number of enterprise applications is not easy. It covers not only enterprise applications but also integrations. In order to carry information from one application to another, integration can exchange information between different applications.

The management system in enterprise Web services environment is quite different with traditional enterprise system. It should cover the following features:

- The standards supported by a service
- Services Interoperation
- Policy control
- Integration with existed environment easy
- A uniform management interface

II. Background

There are many standards those define the communication protocol and what agent should have, such as SNMP, DMI, CIM, WBEM, and WSDM. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth. However, using SNMP, the interfaces should be defined during program compiling time. DMI provides management protocol and architecture, and can add management functions dynamically, but it does not define methods to collect information from several computers in a distributed environment. CIM is developed to solve the problems. JMX provides a Java technology framework for building distributed, Web-based, modular and dynamic solutions for managing as well as monitoring devices, application, and service-driven networks. On the contrary, WSDM is a set of management specifications that provides more structures and semantics for a manageability interface than JMX does. WSDM is suitable for heterogeneous

978-0-7695-3687-3/09 $25.00 © 2009 IEEE
DOI 10.1109/NISS.2009.147

environment. In most Web services, it is selected as management standard. Therefore, our proposed management architecture follows WSDM standards.

Since enterprise Web services environment is more and more popular, Web services management has became one of the most features for the software [1], [2], [3]. IBM provides a complete management product, Tivoli [1], which provides API for developing management applications and a rich client that can monitor and control managed resources. Recently, IBM adds WSDM into Tivoli so that it can communicate with other company products easier. Unicenter [2] also contains WSDM which enables different management areas, predominantly performance and security. The closest to our research is AmberPoint [3] Service Level Manager that enables definition, monitoring, and management of custom-made SLAs. There are also some works on web services management using WSDM standard. In [6], an extension of existing Web services-agent integration toolkit WS2JADE for Web services management is presented, especially using WSDM interface. The work in [4] proposes a novel WSRFWSDM and Multi-agent based Distributed System Resource Management Scheme. The work in [6] proposes a novel SOA-based system and network Management Scheme-SOAMS, and then design a novel WSDM-based management middleware model for SOAMS. Besides, research [7] also present a integration framework for building enterprise computing platform In most of above mentioned work, programmers still have to pay great efforts to develop management functions. If an existed service needs to be managed, extra code is needed and the service has to be rewritten. Besides, most of them focus on single Web services management. However, one business operation usually needs cooperation of several services. For the reason, a complete management system should control the whole enterprise services environment, not only single service.

III. Management Architecture

A. Overview

To provide flexible and extendable management architecture in the enterprise web services environment, the agent-based architecture is proposed. Agent can collect manageable resources information. However, those resources still have to implement the management functions. It will affect the existed services program code. To solve the problem, hook technology is used. It can inject the manage code into the existed services without any code modification. The program logics include service information collection and WSDM end-point implementation. Therefore, it can reduce the effort to modify the program code. After the agent gets the information, it will provide the information to the management applications using the predefined protocols. Then, system administrator can monitor and control the whole system.

3.2 Architecture

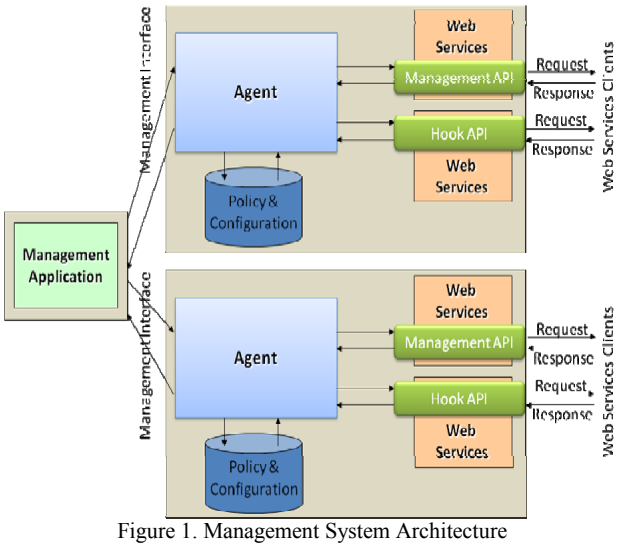

Figure 1. Management System Architecture

The management system architecture is showed as Figure1. For each service, it will integrate Management API / Hook API. They are responsibility for monitoring Web services status and provide necessary information to agent. The APIs can monitor the whole process from invoking service to returning result. When a request is sent to the services, the APIs captures the request first. They will process the request and record information. Part of the content in the request message has identification added by another Management API/Hook API in the management system. The content is useful to monitor and control each service. Those APIs will also process the content in a request message. For the reason, if a service issues requests to another service, the APIs will capture the request first and add monitor information into the message. Therefore, all service information in the management system can be captured.

Agent is deployed between Management applications and services. It reads the system policy and configuration from the database, gets service status from Management API/Hook API, and communicates with Management applications. Because Management API/Hook API exports both get and set operation for services, Agent can get services status quickly and set parameter to services directly. The management information will send to Management applications when it requests. Management applications can also send service parameters to Agent to adjust service behavior. For each host, there is only one Agent

deployed. In other words, each service would not have the management interfaces. They are handled by Agent. Therefore, Agent will provide communication interfaces. Management API/Hook API can communication through the interfaces. Single management interfaces on Agent can reuse the similar management functions and coordinate resources needed for management purposes. For example, if Agent is not here, Management application has to build several connections for each service. However, with Agent, the number of connection for a host is only one. Agent helps to reduce Web services implementation effort and improve system performance.

For each Agent, Policy & Configuration database is deployed. It is the storage that Agent can read and save information. Because the policy or configuration is complex and consisted by many parameters, it is not easy that Management applications set those parameters one by one, especially when parameters may affect each other and one error would cause service crash. Each Agent can save those parameters by themselves. If necessary, those settings can be retrieved quickly. It guarantees system stability and save management effort.

Before describing detail design for every component, a characteristic of the enterprise web services is introduced. Each service contains several operations. They can be separated as Business Services and Technical Operation.

B. Business Services & Technical Operations

Every service in enterprise is built for business purpose. It can solve specified business problems. For example, a ProcessOrder operation in a service is used to handle order. Those operations can be mapped to business purposes. However, it is not enough if there are only business services. In distributed environment, some technical operations are also provided to support specified technical requirements. Most Web services export WSDL for service consumer to retrieve service description so that the consumer can know how to invoke the service. For the reason, technical operations are also important. A server that can perform business services well relay on those technical operations.

Generally, Business Services fulfill business requirements and those requirements are usually different. Therefore, only the different business requirements are implemented. If there are similar requirements at two systems, only one will implement and another will reuse it.

On the other hand, Technical Operations at different services are similar. For example, if one service needs transaction support, it has to implement those technical operations and protocols. Because those requirements are similar, a reused program code can be created and all systems will relay on it.

Besides, it is not easy to plug the management technical operations into an existed service if they are not existed in a service. The general steps to add the management functions are showed as Figure2. First, a WDSL file is necessary. It can generate the stub and skeleton for serve side and client side, respectively. Then, developers can develop the service using the generated code. Finally, the service has to be deployed at the server. The procedure takes a lot of time. For most managers, they do not accept to spend resource when the Business Service does not change. For the reason, an efficient way is necessary.

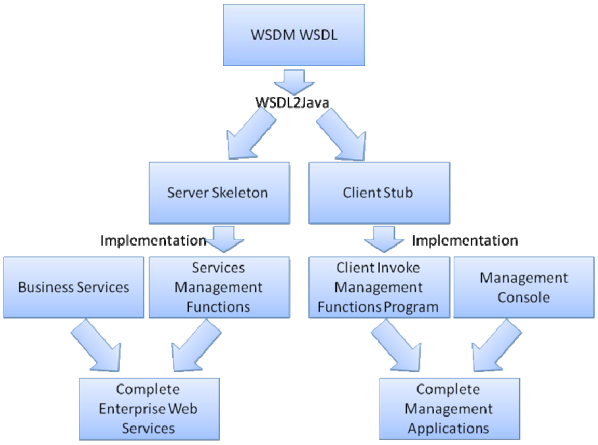

Figure2. Implementation Flow for WSDM

C. Management API

To access and configure services are the most fundamental functions in a management system. It is also the most important part. However, it usually costs a lot of time to develop those information access and system configuration functions. Although there are many management products, most of them provide APIs that focus on management protocol and management applications. Programmers still have to follow the management architecture and APIs carefully when developing a service. Otherwise, the service cannot be managed.

In enterprise Web services environment, it is necessary to define managed resources and provide concise APIs for developers. Developers, then, can write the management code using those APIs. For the reason, an approach that can generate interfaces and add management capabilities into services is proposed. They are Management / Hook API. Management API provides functionalities those will get the status of managed resource. Programmers can export service status to agent using this APIs. Besides, if the web services status can be specified precisely in program code, the Hook API can be used. It can access program code of the service so that the management information can be got without code modification. Management API focuses on how to export managed resource status

to the management system. It contains management capabilities and management resource. Management capabilities indicate what kind of capabilities the resources are. One resource may contain several management capabilities. Their relationship can be showed as Figure 3. Those capabilities and resources are associated with a physical hardware or resources. Those capabilities contain management information and operations those describe the status and operations of the back-end resources. The managed resources expose the capabilities to be used in management for Agent.

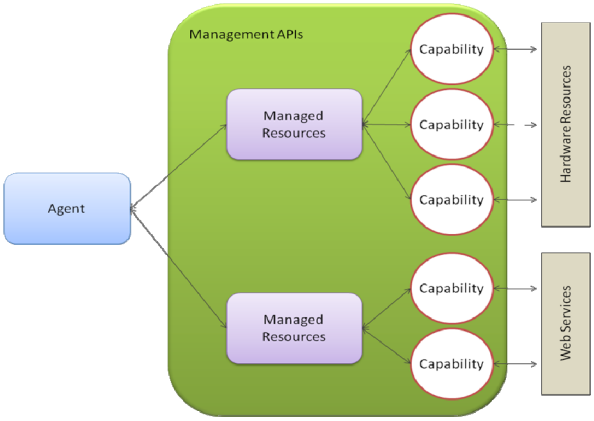

Figure 3. Managed Resources and Capability Object in Management API

The capability contains data and operations for a resource. It is atomic unit in the management system and represents the capabilities of a resource. Single capability shows one kind of management information for a resource. For the reason, a resource will have several capabilities.

D. Agent Design

Agent in the management system is the bridge between Management applications and capability implementation class. It has the following features:

1. Web services interface: Agent exposes every Managed Resource as Web services.
2. Basic technical management information: Agent can provide basic metric information.
3. Service configuration: Agent can read configuration from the registry and send those configuration to the corresponding services.
4. Security: Agent has the Web services interface that can be accessed by anyone. Therefore, security mechanism is necessary.

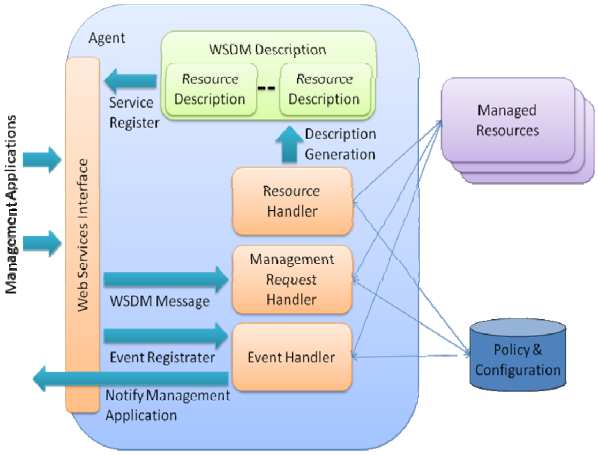

Figure 4. Agent Internal Components

IV. Web Services Process Management

The software as a service model can overcome many limitations that constrain traditional software use, deployment and evolution[8]. Web services architecture constructs such service environment and provides a suitable technical foundation for making business processes accessible within an enterprise because of the standard interfaces as well as communication protocols. The architecture focuses on separating the possession and ownership of software from its use. Delivering software's functionality as a set of distributed services. Therefore, the business requirements can be fulfilled only when composing those services. Those composing Web services make up business processes that accomplish business requirements. For the reason, managing enterprise systems does not only monitor service status but also guarantee Web services process execution well.

A. Web Services Status Monitor

To perform a Web services process, it is necessary that every service can accomplish requests. Each Web service should accept the request and response correctly. Otherwise, the whole process may encounter errors. For the reason, system administrator should monitor service status and take actions when there is something wrong.

B. Service Composition

The importance of service composition has been widely recognized in the distributed research community due to its high flexibility in allowing development of customized applications. Enterprise web services cannot exist along. They most co-operate with other services to accomplish business and technical needs. However, the two kinds of needs have different concerns.

C. Services Composition for Business Needs

Web services are built to support business requirements. There are many concerns about services composition. For example, the selected service for composition should consider multiple criteria (e.g., duration, reliability)[9]. Some researchers define semantic Web services to help service selection [10][11]. After studying those researches, Web services composition should satisfy business requirements, including business functionalities and service quality. Although Web services architecture has flexibility to select suitable services, it does not define measure methods or standards. In other words, service clients have to handle and record the quality information by themselves.

Enterprise IT systems should improve and adjust by current target and status [12][13]. Nowadays, more and more corporations build not only internal systems (e.g., ERP) but also external systems (e.g., CRM, SCM, and SRM). Those systems can earn more business opportunities or reduce cost. If IT systems can adjust for the change quickly, corporations can make the best response. Web services architecture has proven the adjust solution. The standard interfaces make the communication between heterogeneous systems possible and dynamically service composition. However, it is hard to find out the service bottle without proper tracing mechanism. When a request is issued into the Web services environment, the system should know all services involved and their order. Therefore, the system can know the detail information from management framework of the services and adjust services as they need.

D. Message Flow Oriented Management

Message flow represents how the message is processed. Figure5 shows a sample invocation among Web services. There are many invocation happened at the same time in enterprise. It is not to find out a bottle bottleneck or problems during service invocation. In current Web services model, service clients only concern about the next service provider. Those clients only recognize that the next service has problem but they cannot provide the message source to adjust services. For the reason, to join Web services together with message flow can provide useful information to manage them.

E. Message Flow Tracking

Because of service composition dynamically, it is necessary to trace message flow to prevent lost or queued. On the other hand, the tracing mechanism should not impact overall system performance and cause many design as well as implementation efforts. For the reason, an automatic message flow tracing mechanism with few performance influenced is required. The message flow tracing in the management system is illustrated as three steps:

(1) Message logging
(2) Information collection
(3) Message flow analysis

As the message flow in Figure5, Service A, C, D will get service request in turn. Without message flow tracing, it is hard to find the relation between those services. However, in the management system, Requestor Handler of every service will get the ID from message and Invocation Handler will put the ID into the message sent to the next services. The two handlers only record the ID and time. Then, Agent can get the information from registry and provide a Web services for management application. Management application gets the message flow tracing information from all Agents and makes up the flow. Although the final process costs a lot time, it is not done at the services side but a standalone system. Therefore, the architecture does not affect the overall system performance.

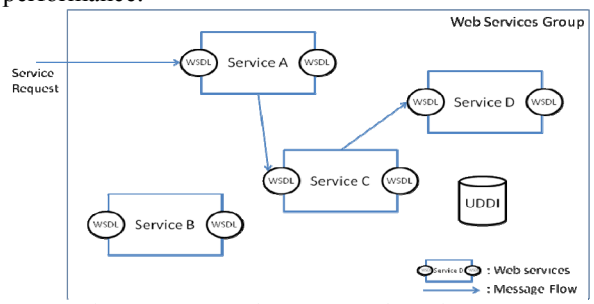

Figure 5. Message Flow among Web Services

F. Dynamically Service Selection

With the completeness of Web services environment in a corporation, the service requestor will have the chance to choose suitable service. A service is unavailable for many reasons, like maintenance, server loading high, hardware error, and software fault. Under such cases, service requestor has to issue request to another service. In current enterprise Web services architecture, it is not usually to look up UDDI before invocation for performance issues. For the reason, a adjustable solution for service selection can improve service quality.

Invocation Handler of Hook API can help to change service from one to another and redirect message to that service. Because the handler intercept the process that service requestor sent message to the service, it can re-arrange the message destination. Invocation Handler accepts administrator's configuration to create a Web services PortType mapping table with XSLT translation document. When a invocation message is sent to the original PortType field, it will redirect the message to the new Port type. At the same time, the message will be transformed based on XSLT if necessary.

G. Exception and Error Handling

If there is an exception or error during service invocation, the exception or error handling procedure will handle the exception or error. Usually, the procedure will record the problems and notify related people to eliminate it. It is the same in the Web service environment. The procedure will log the error and send a notification to the management architecture. Besides, if there are exceptions or error handling message flows, the flows will startup to deal with problems.

Exception and error handling flows in the management architecture are treated as special service selection cases. The different is that the flows for exception or error start when error occurred, but flows for service selection is used when the destination service PortType is match with one entry in PotyType mapping table. Invocation Handler will invoke the service at the first of exception or error handling flow and the mark the message as error in Message ID.

V. Conclusion

The management system who proposed in this research not only focuses on single service but also monitor overall enterprise web services environment. It creates development and run-time architecture to manage a service. WSDM standards were followed so that applications can manage them directly. Developing those management functions doesn't have to understand all WSDM specifications. Programmers only write the management code. The management system will expose them as Web services automatically.

Based on the architecture, the message flow in the services environment can be monitored. Managers can know status for every message so that they can adjust business process or improve system performance to enhance business operations.

With the management system, the enterprise Web services environment will have a complete management mechanism. Not only single Web service is managed, but also all message flow is under controlled. The environment will give corporate better support.

Acknowledgment

This study is conducted under the "Service-Oriented-Machine Open Platform Research and Development Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy of the Republic of China

This study is also partially supported by grant of Science Education Department of the National Science Council of the Republic of China. under NSC 96-2520-S-007-MY3

References

[1] IBM Tivoli Software, http://www-306.ibm.com/software/tivoli/

[2] CA Unicenter product, http://www.ca.com/us/products/default.aspx

[3] AmberPoint Comprehensive Web Services Management, http://www.amberpoint.com/

[4] ZhiHui Lv ShiYong Zhang Jie Wu WeiMing Fu YiPing Zhong , "A Novel WSRF and Multi-Agent based Distributed System Resource Management Scheme", The Sixth International Conference on Grid and Cooperative Computing(GCC 2007)

[5] ZhiHui Lu, Yong Li, Jie Wu, ShiYong Zhang, YiPing Zhong, "SOAMS: A Novel SOA-based System and Network Management Model and Scheme", IEEE International Conference on Services Computing, 2008. SCC '08.

[6] Xuan T. N.,Kowalczyk, R., "Enabling agent-based management of Web services with WS2JADE", Fifth International Conference on Quality Software, 2005.

[7] Ruey-Shyang Wu, Fengyi Lin, Shyan-Ming Yuan, Kai-Chih Liang, A reference model and integration framework for building enterprise computing platform, International Journal of Technology Management(SSCI), Vol. 38, No.4 pp. 439 – 462,2007

[8] M. Turner, D. Budgen, and P. Brereton, "Turning Software into a Service," Computer, vol. 36, no. 10, Oct. 2003, pp. 38-44.

[9] Liangzhao Zeng , Boualem Benatallah , Marlon Dumas , Jayant Kalagnanam , Quan Z. Sheng, "Quality driven web services composition", Proceedings of the 12th international conference on World Wide Web, May 20-24, 2003, Budapest, Hungary

[10] Paolucci, M., Kawamura, T., Payne, T.R. and Sycara, K. "Importing the Semantic Web in UDDI." Proceedings of E-Services and the Semantic Web Workshop, 2002.

[11] Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D. , McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T., Sycara, K., "DAML-S: Web Service Description for the Semantic Web," Proc. Int'l Semantic Web Conf. (ISWC), LNCS 2342, Springer Verlag, 2002, pp. 348-363.

[12] L Aversano, M Tortorella, "An assessment strategy for identifying legacy system evolution requirements in eBusiness context", Journal of Software Maintenance and Evolution, Vol. 16, NO. 4, 2004, pp.255-276

[13] Harry M. Sneed, "Integrating legacy Software into a Service oriented Architecture," csmr, pp. 3-14, Conference on Software Maintenance and Reengineering (CSMR'06), 2006