WS-RM: Resource Management Using Web Service

Yu Wu, Zhihui Lv
School of Computer Science and Technology
Fudan University, Shanghai, China, 200433
{yuwu, lzh}@fudan.edu.cn

Yuan Yuan, Chaoyang Wang
China Electronics Standardization Institute
Beijing, China, 100007
{yuanyuan, wangcy}@cesi.ac.cn

***Abstract*—We have been participated in the early research work of using Web service for system management and its standardization work. In this paper, we first review the concept of using Web service for management and compare two main industry standardization efforts - WSDM and WS-Management. We present WS-RM, our resource management model, its implementation, the testbed and management scenarios. Then, we present several selected gap items that are identified between real-world management requirements and current standard as well as our proposed solutions. In the end, we discuss some future work into cloud computing management and using Web service for management integration.**

I. Introduction

The enterprize IT environment is becoming extremely distributed and heterogeneous today. Variety of hardware and software have been deployed over the time and across different domains or physical locations to support business applications, some of which could be legacy system while others are based on the very update technologies. The success of deployment and operation largely depend on the system administrator's ability of efficient management so as to secure the business continuity and availability. The term "system manageability" represents a wide range of technologies that enable remote system access and control in both in-service (OS-present) and out-service (OS-absent) environments. These technologies are primarily focused on minimizing on-site IT maintenance, maximizing system availability and performance to the local user, maximizing remote visibility of (and access to) local systems by IT managers [1].

Though a broad set of tools and technologies are available for remote management resources (also known as client systems in some literature) that are powered on (in-band) with a responsive operating system (in-service), most of these tools are tightly coupled with the managed resources through their own proprietary interfaces, e.g., SNMP, WMI, JMX and etc. Individual management operations and monitoring metrics have to be defined for each type of resources. When the clients are powered off (out-of-band) or out-of-service, the list of management options decreases dramatically and most the options are based on proprietary technologies limited to client systems from certain vendors.

Except for the high cost of higher licensing and upgrade fees, relying on proprietary management tools or proprietary protocols can reduce the flexibility to quickly respond to the changing needs of different user groups in an IT organization. When computing technologies rapidly evolve and new devices appear, different tool sets have to be developed to manage these clients. As the new proprietary solution are not likely to interoperate well with the existing ones, the end result is increased management complexity and higher risk of jeopardizing the stability.

The wide adoption of Web service has introduced a new approach to resource management, a more consistent way to access various resources. By leveraging the existing set of core Web services specifications, the industry-supported Web services architecture, Web service based management could provide a consistent method for remote management of devices, from traditional distributed systems to new SOA-based applications.

The realization that IT systems are becoming more and more distributed connected and the ever increasing requirement of a common way to access resources and exchange management information across the entire IT infrastructure are the primary motivations behind the adoption of Web service for system management. By providing a uniform and network-friendly interface, Web service is expected to make it convenient to integrate management tasks vertically (between multiple layers of the IT stack) and horizontally (across distributed applications and products of different vendors). Such demand leads to efforts to standardize a vendor-neutral, platform independent protocol for redeveloping management tools.

The paper is organized as follows. In Section II, we first introduce the concept of using Web service for management and resource modeling. In Section III, we first compare two efforts on the standardization, WSDM and WS-Management. Then we compare the de facto industry standard, WS-Management with several other non Web service based protocols, IPMI and ASF. In Section IV, we propose WS-RM, our WS-Management based resource management model, and its evaluation work using a three tier web application system and two common scenarios in Section V. In Section VI, we present several identified gap items between real-world management requirements and the specification. We also discuss their solutions. In Section VII, we present the related work and then conclude the paper in Section VIII.

II. Concept of Using Web Service for Management

The rapid development of Web service and its key characteristics provide unique advantages for system management. Web service uses open standards and protocols which are widely

978-0-7695-4756-5/12 $26.00 © 2012 IEEE
DOI 10.1109/SERVICES.2012.51

supported by different vendors, providing the interoperability between management applications and various software applications running on separate platforms without constraining the component implementation. Any management vendor can use these Web services interfaces, reducing the amount of custom support or development required. Its good interoperability and loosely-coupled nature makes it a good tool to exchange management information and to promote integration between management applications and various kinds of managed resources in today's distributed and heterogenous environment.

Figure 1 illustrates the conceptual design. Managed resources (or client systems in some literature) expose its manageability capability, which is a composable set of properties, operations, events, metadata, and other semantics that supports management task through its management interface as Web service access point. There are manageability capabilities that are globally applicable to many resource types, e.g., state management as well as that are unique to a single-managed resource, for example the Java virtual machine (JVM) manageability capability only applies to JVM-managed resources. Any SOAP compliant management clients (or managers in some literature) can access the resource through access point and exchange information using XML-based message.

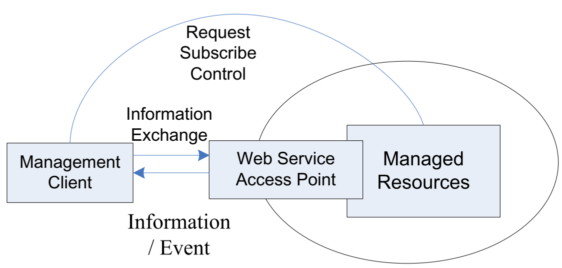

Fig. 1. Concept of using web service for management

The managed resource is modeled using modeling language, such as CIM (Common Information Model). CIM is an object oriented modeling language that can be used to define the actual structure of resources, containing the definitions of the entities of managed resources, such as their properties, their permissible behaviors, and the operations that may be performed upon them. CIM is composed of a specification, which defines the details for integration with other management models and a schema, which provides the actual model descriptions. In addition, CIM profiles are collections of classes needed to implement a specific management feature, such as resources that support power state management could be expected to support the Power State Management Profile. This profile would define all the individual CIM classes necessary to conduct power management operations.

III. Development of Industry Standards

In this section, we first introduce two main efforts on industry standards and then compare WS-Management, the de facto industry standard that has been widely supported with other non Web service based protocols.

A. WSDM and WS-Management

Over the time, there are two main efforts on exploring the usage of Web service based management solution and its standardization work. First is WSDM (Web Services Distributed Management) from OASIS in 2005, which consists of two standards, MUWS [3] (Management Using Web Services, a protocol) and MOWS [2] (Management Of Web Services, a model). WSDM makes use of WSRF [4] (Web Service Resource Framework) which is started as an attempt to align the GGF/OGSI approach to resource access with the IT management approach. The purpose of the WSRF is to define a generic and open framework for modeling and accessing stateful resources using Web services. Therefor, WSDM could be regarded as an extend of grid work into the management field. There are a few WSDM implementations, e.g., Apache Muse, a Java-based implementation and IBM WSDM development kit, but it did not achieve real adoption and the OASIS technical committee was closed after a few updates.

The WS-Management specification is first initiated and developed by a consortium formed by Sun, Microsoft, NEC, Intel, Dell,and other companies. Later. the specification was submitted to DMTF (Distributed Management Task Force) in 2005 and released as a preliminary standardization effort in 2006. Its most update version is v1.1 and is released in 2010 [12]. It is based on a series of general SOAP-based protocols, including WS-Transfer [10] for accessing XML representations of Web service-based resources, WS-Enumeration [5] for enumerating a sequence of XML elements that is suitable for traversing logs, message queues, or other linear information models, WS-Eventing which allows Web services to subscribe to or accept subscriptions for event notification messages and WS-ResourceTransfer [10] which defines extensions to WS-Transfer with initial design focusing on management resource access.

Figure 3 illustrates WS-Management protocol architecture [7]. The WS-Management standard identifies a core set of Web service specifications and usage requirements to expose a common set of operations, such as PUT, GET, DELETE and CREATE that are central to systems management for information retrieval and exchange. It also provides a mechanism, supported by WS-Eventing for users to subscribe to get alarm indication, for example when the CPU usage has exceeded the pre-defined threshold, a notification is sent to the management system.

WS-Management is similar to WSDM in many ways and the main difference lies in their focused perspective [27]. WSDM provides a richer solution which aims at managing distributed systems and SOA applications that compose of resources of very dissimilar types. WS-Management is a more lightweight specification which addresses more on communication protocol and reuses the modeling work from other specific standards, such as CIM [6], JMX [9], IPMI [16], etc.

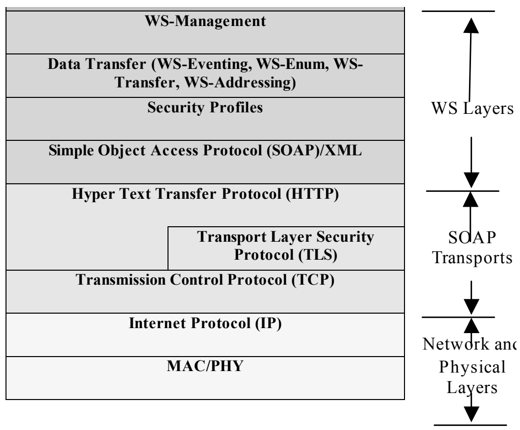

Fig. 2. WS-Management protocol stack

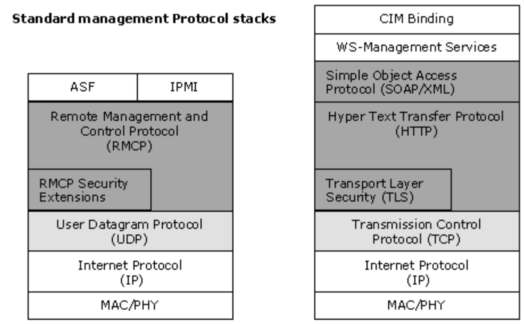

Fig. 3. Comparison between WS-Management and ASF, IPMI

WS-Management defines a set of foundation manageability capabilities for basic management concepts and tasks. New or domain-specific capabilities can extend existing foundational capabilities as appropriate. For example, considering the different properties of managed resources and management requirements, there are some other initiatives extending WS-Management into different domains, such as SMASH (System Management Architecture for Server Hardware) [8], SVPC (System Virtualization, Partitioning and Clustering) [14] and DASH (Desktop and mobile Management Architecture for System Hardware) [7].

B. WS-Management and ASF, IPMI

Before WSDM and WS-Management, there are existing industry standards enable management systems to exchange management information such as ASF (Alert Standard Format) and IPMI (Intelligent Platform Management Interface). The ASF standard was developed primarily to support remote management of client systems in OS-absent (in-service) environments, while IPMI was developed to instrument server hardware. Both ASF and IPMI are based on RMCP (Remote Management Control Protocol) which is a UDP based networking protocol. In contrast, WS-Management is a multi-vendor initiative to develop a standard protocol for remote management. It is based on standard internet based protocols, such as SOAP/XML, and Web services and is developed as a management protocol for all IT infrastructures, which is not limited to just desktop management or server instrumentation. Figure 3 illustrates the protocol stack comparison between WS-Management and ASF, IPMI [15].

IV. WS-RM MODEL FOR RESOURCE MANAGEMENT

Considering the large existence of proprietary management protocols and new arrival of Web service enabled devices, we developed a resource management model, WS-RM (web service based resource management model). WS-RM adopts Web service for management in two approaches, native and hybrid and consists of three layers, agent layer, gateway layer and manager layer. The model itself is independent of the management standard it adopts and is general enough to suit in most real-world management scenarios. We combine the model with WS-Management specification and illustrate its architecture in Figure 4.

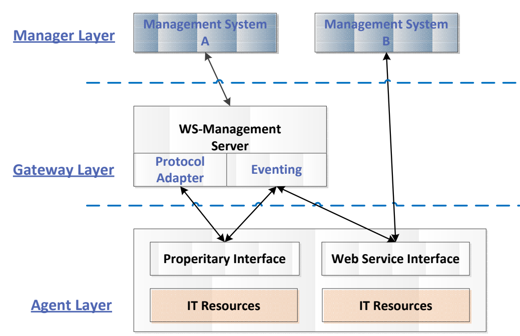

Fig. 4. WS-RM model architecture overview

The agent layer usually locates on the managed resources, such as a physical server, an application server or a chip set. It is the interface which exposes the resource management capability that is defined in the management model of its type. The management model consists of the attributes and operations available for system management. For example, any WS-Management compliant manageable resources must support four operations, PUT, GET, DELETE and CREATE for information retrieval. Also customized operations can be defined and added.

If the resources have already exposed Web service based interface, such as WinRM of Windows Vista and Server 2003, management clients could talk directly to agent layer to get monitoring data or execute operations using SOAP-compliant messages. This is referred to as ***native*** approach. For the resource using its own protocols, various kinds of adapters are developed to map these proprietary protocols to WS-Management, such as JSR262 [11] for JMX [9]. This adapter-based approach is referred to as ***hybrid*** approach. These adapters are usually hosted in WS-Management server in the gateway layer and are responsible for the communication with agent layer. The Gateway layer sometimes are coupled with agent layer on the managed objects or deployed on another physical server. One WS-Management server could be responsible for a number of managed resources of different physical or logical locations. Manager layer is where high level management software or clients initiate SOAP-based messages to retrieve management information, execute operations and perform management integration.

WS-RM provides a flexible way to access both existing protocols and new Web service protocol enabled resources, promoting for better interoperability between managed resources and management system. Figure 5 illustrates the architecture of integration WS-Management with IPMI for server management in today's hardware management [23], demonstrating both native and hybrid approaches.

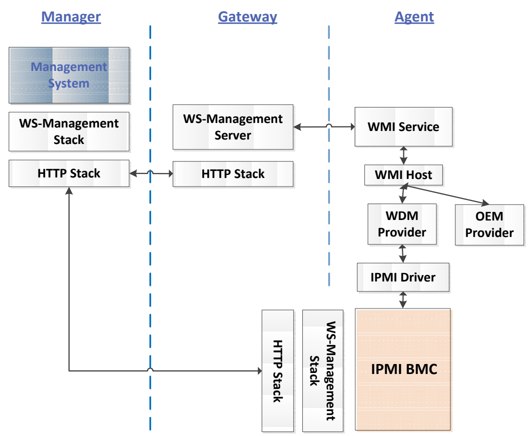

Fig. 5. Integration of WS-Management with IPMI

V. Implementation and Evaluation of WS-RM

The implementation and evaluation work of WS-RM model helps us to better understand how Web service can help to ease the burden of system management in real world scenarios.

A. Testbed

We construct a three-tier web application infrastructure as our testbed and illustrate its topology in Figure 6. Web server 1 and web server 2 are virtual machines on a single DELL 1900 and web server 3 is a single physical server. We configure round-robbin mechanism for Tomcat cluster to dispatch requests. A single primitive service, book search, has been deployed. User invokes the service to query the detailed book information. Web server then parses the request, retrieves the result from back-end database and send back to the user. The service is hosted in Tomcat server which is running on Windows server platform. User requests are generated from two JMeter clients deployed at two individual desktops.

There are three kinds of managed resources in our target system and we list the managed resources, properties and their definitions in Table I.

In our implementation, For agent layer, we use JMX for Tomcat and WinRM for Windows platform management. JMX

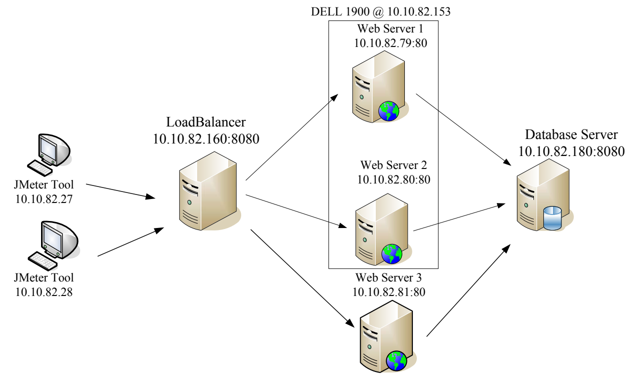

Fig. 6. Topology of three-tier web application testbed

is the default management protocol for Tomcat and a rich set of Mbean have been integrated to expose different run time performance metrics, such as NumberOfActiveRequest and NumberOfRequest. In gateway layer, we develop an adapter to transform JMX to WS-Management with reference to JSR 262. JSR 262 (Web Services Connector for Java Management Extensions (JMX) Agents) defines a mechanism for the JMX Remote API that uses Web Services to make JMX instrumentation available remotely. Now its reference implementation, Ws-jmx-connector is available online. And we choose the open source project Wiseman, a java implementation of WS-Management specification as the WS-Management server. This project is initiated by HP, SUN and now maintained at *dev.jave.net* community.

We implement two common management functions, retrieving monitoring data and receiving alarm events. During runtime, the management system keeps monitoring its current status, either through pull the information from the managed resources or such information could be submitted from clients. For a specific metric, administrator may set its max or min threshold and subscribe to the WS-Management server for abnormal alerts. For example, when the value of the metric exceeds the pre-set max threshold, a notification is sent. On observing this event, system administrator can execute corresponding operations, such as add a server to the cluster to increase the capacity of the system to cope with the climbing workload.

B. Evaluation

In our experiment, we simulate two different management scenarios and each experiment lasts for about one hour and is repeated several times. All management data is retrieved every ten seconds and we use the average value of last five data points to plot the diagram for display.

First is the normal scenario during which target system workload is within a normal range and management information is retrieved at the pre-set frequency and then displayed. The experiment is to evaluate the monitoring function. In this scenario, we first start three web servers and then start

TABLE I
PROPERTIES OF MANAGED RESOURCES

Resource	Property	Description
Book Search Service	NumberOfActiveRequests NumberOfRequests	sum of the number of current requests of all web servers sum of total number of requests that have been processed of each web server
Web server	NumberOfActiveRequests NumberOfRequests	the number of the requests that are being processed total number of requests that have been processed
Hardware server	CPU usage Memory usage	current usage of server cpu current usage of server memory

JMeter to generate requests to stimulate workload for the target system. The requests are dispatched through load balancer onto the three web servers evenly. The workload from the client remains unchanged during the experiment. Second is the overload scenario, during which the target system is overloaded. In this scenario, when receiving the alarm, administrator execute an operation to add a new server into the cluster. The experiment is to evaluate the eventing and operation function.

Through evaluation, WS-RM is proved to be effective for daily scenarios, but we also identify several gap items while trying to map management requirements. These gap items include both architecture design and specification details, which are discussed in the next section.

VI. GAP ANALYSIS

Goal of the evaluation work is to identify the gaps between real-world management requirements and the specification. The gap items are first proposed, identified if they are within the current scope of our the specification and then evaluated in our testbed. During this analysis work, we have intensive discussion with China Web Service Management Working Group members, which is responsible for the standardization work under the supervision of CESI. We also had a discussion with experts from OASIS and DMTF. Some of gap items are accepted and reflected in the newer version of WS-Management specification, while some are transferred to other working group, such as WS-Eventing, WS-Resource, etc. In this section, we present several selected gap items and our proposed solutions.

Discovery of management capability The first step of resource management life cycle is often the discovery of its presence and the management capabilities, including properties and operations available. When a new resource is connected, it would be better for management client to discover the capabilities that have exposed. Though capability definition is part of management model, capability discovery is a common scenario and shall be addressed in the specification. For example, when the new resource is detected, management client can send a request to enumerate its properties and operations (method definition and its parameter). Capability discovery is also useful when resources are updated or modified, so as to reduce the burden on administrator to manually trace the change.

Destruction of a resource instance The last step of resource management life cycle is destruction of a resource instance, either when the resource is removed or temporarily unavailable. As the number of managed resources is usually large in today's IT environment, we propose time-based, scheduled destruction of a resource instance. When a resource is discovered, management software can create a corresponding instance for management and specify an expire date, after which recreation is required. The expiration date could be initiated according to the resource type.

Subscription status Management client may subscribe to WS-Management server to receive alarm messages. But how client can detect current status of subscription, e.g., still valid or not. There are several optional implementation choices in the specification, such as subscriptionEnd, getStatus or heartbeat. The working group members suggest that a simple re-subscribe with the UUID that is generated in the first subscription can simplify this task. Based on the reply message, such as error information (duplicated subscription) or success message, it is more easier for client to determine current status of subscription and reduce message interactions.

Number of connection retry A WS-Management server should have the policy about the maximum number of connection retry, minimum interval of two consecutive retries when first notification fails. Also, the server should be able to constrain or adjust those parameters which the client can specify in the Subscribe request based on the resource type, resource constrain, etc. On the other hand, client should be able to know the policy supported by the server side and the parameter range so as to define proper value in subscription.

Maximum number of subscriptions The maximum number of the subscriptions a WS-Management server can support is limited, especially for embedded systems. In this situation, management client should want to create only necessary subscriptions. Therefore, maximum number of subscriptions server side can support and the number of existing ones should be made available to client. We propose to keep a counter either at client side or on the server side and whenever the number exceeds the limit, an error message shall be returned.

Also, we have discussed the different perspective of *Protocol and model*. In system management, besides the protocol, there is another equal important part, resource modeling. While the WS-Management specification defines most basic and common operations that can be performed on resources, it does not define what an actual resource is. It is the role of resource modeling, e.g., CIM. Though this part of work is not addressed by WS-Management and it is not supposed to, the integration of the two is of great concern for both hardware designers and software developers.

Interoperability between existing resource model and WS-Management specification becomes one key concern for hardware/software vendors when considering whether to upgrade their existing firmware. In Web service world, XML is the standard common data format, which is easy for machine to interpret and process. But in system management, current (proprietary) management protocols and their resource models have its root in meeting different requirements of that individual domain, e.g., desktop servers, laptops, mobile device and etc. Because of this, they usually have their own data format and model characteristics. For example, JMX is a more natural choice for JAVA based application management, but the way it serializes data is bounded to JAVA platform. What's more, from our own implementation work as described in Section V, over 50% effort is spent on integrating with existing management protocols and resource models, e.g., JMX. The availability of adapters for different types of resources, such as JSR 262, WS-Management/CIM binding is necessary to reduce the burden on developers and help to realize full potential of using Web service for management.

VII. Related Work

Over the years, there have been efforts exploring using Web service to build management solution in heterogeneous environment. In [26], [28], the author proposes a Web services-based management model for the distributed system and network resource. In [17], the author proposes an architecture that uses Web services and automatic Web services composition as a complementary technique for policy refinement for network control. In [25], Neisse et al propose a gateway based approach to integrate SNMP devices with Web services and verify its feasibility of using Web services closer to the network devices interface.

A more detail comparison between WSDM and WS-Management specification about their focuses and protocol stacks can be found in our previous work [27]. In [24], the author presents the results of a set of experiments carried out in order to compare the specifications MUWS and WS-management against the de facto network management standard (SNMP). The performance metrics investigated are network usage, response time, and CPU usage. In [19], [21], [22], we have explored building management middleware using WS-Management. Similar work is also reported in [18], which cites industry best practices when using WS-Management to bind services and other manageable entities in the service-based architecture. In [20], we briefly report three use cases of using WS-Management for system management based on MAPE (monitor, analysis, plan and execute) management cycle.

To the best of our knowledge, this is the first work focusing on the real-world experiences, its evaluation and best practice of using Web service (WS-Management specification). We also introduce the standardization work in China. The use cases and gap items that have been identified come from real world management scenarios and user demand from both consumers and providers, thus they serve as best possible justification for any standard work. Also, we summarize the lessons learned in standardization process and our future work. We hope it would provide some hints for the development of the technology itself as well as the standard.

VIII. Conclusion and Future Work

In this paper, we review the initiative and motivation of using Web service for management and compare two main industry standards - WSDM and WS-Management. Then we propose our resource management model, WS-RM and its evaluation using a three tier web application in two different management scenarios. As the important part of this paper, we summarize the gap items that are identified through a careful comparison of WS-Management specification and real-world management requirements. We discuss the different technical requirements of management protocol and the model as well as the demand for more adaptation on integration of WS-Management with other existing resource models.

With the inclusion in a large number of manageable products with long development cycles (servers, devices, operating systems) ensures that WS-Management has been accepted and adopted as a protocol for resource manageability and will get continuous support and update. However, the potential of using WS-Management for management integration is not yet fully explored. For now, WS-Management is mainly adopted as a protocol for resource manageability, not yet for management integration. Though the difference between manageability and management integration is not always clear-cut, the truth is that there are not many scenarios that urge the adoption for integration, even if SOAP based interaction is also a good candidate for integration. However, the rise of cloud computing has given rise to the paradigm of nearly infinite elastic scale computing, where on-demand resource provisioning could expand more flexible. Meanwhile, it also proposed a more severe challenging on managing the dynamic sharing environment across different management domains and thus requires a level of abstraction over manageability to provide an integrated system view of IT resources. However, current design of WS-Management is not ready for this new task, not by a long shot.

What's more, many international standards have been proposed for specifying resources as well as how they can be efficiently/securely/reliably managed, which is also the major concern for the community of emerging Cloud Computing (CC). In the future, we would like to expand our work to system management in cloud based environment (Cloud Computing), focusing more on management integration. We hope our work would provide a summary on this topic and some insights for interested audience when aligning efforts of standardization with real-world adoption and implementation.

IX. Acknowledgement

During our research work and standardization process, the working group had a lot of communication with experts from

OASIS, DMTF, W3C, CESI and open source development communities, e.g., WS-Man and openwsman. We truly appreciate all the technical feedback and comments. And we are happy to make our own contribution (source code of some reference implementation as well as gap items) to the development of international specification. The warm welcome and nice host we had for the face-to-face meeting with DMTF WS-Management working group members is an interesting experience. The journey of our joint work is full of fun and we are looking forward to future collaboration.

REFERENCES

[1] Alert standard format (asf), specification, v1.3. 2001.06.

[2] Web services distributed management: Management of web services (mows 1.0) part 2. Organization for the Advancement of Structured Information Standards (OASIS), Standard,, March 2005.

[3] Web services distributed management: Management using web services (muws 1.0) part 2. Organization for the Advancement of Structured Information Standards (OASIS), Standard,, March 2005.

[4] Web services resource framework v1.2. Organization for the Advancement of Structured Information Standards (OASIS), Standard, 2006.

[5] Web services enumeration (ws-enumeration). World Wide Web Consortium (W3C), Specification, 2006.3.

[6] Common information model specification. Desktop Management Task Force (DMTF), 2007.

[7] Desktop and mobile architecture for system hardware initiative specification. Desktop Management Task Force (DMTF), 2007.

[8] Systems management architecture for server hardware (smash) specification. Desktop Management Task Force (DMTF), 2007.

[9] Jsr 255: Javatm management extensions (jmxtm) specification. Feb 2008.

[10] Web services resource transfer (ws-rt). World Wide Web Consortium (W3C), Specification, 2008.12.

[11] Jsr 262: Web services connector for java management extensions (jmx) agents. 2009.

[12] Ws-management: Web service for management v1.1.0. Desktop Management Task Force (DMTF),DSP0226, May 2010.

[13] Cloud computing use caseswhitepaper, v4.0. Cloud Computing Use Cases group, 2010.2.

[14] DMTF. System virtualization, partitioning and clustering specification (svpc). Desktop Management Task Force (DMTF), 2007.

[15] Intel. Ws-management and intel active management technology: A primer. 2008.

[16] Intel, HP, DELL, and etc. Ipmi intelligent platform management interface, specification. 2004.2.

[17] T. Klie and L. Wolf. Autonomic policy-based management using web services. In *CoNEXT '06: Proceedings of the 2006 ACM CoNEXT conference*, pages 1–2, New York, NY, USA, 2006. ACM.

[18] M. Kuhr and D. Hamilton. Building executable service-oriented architectures with the ws-management specification. In *SpringSim '08: Proceedings of the 2008 Spring simulation multiconference*, pages 315–324, San Diego, CA, USA, 2008. Society for Computer Simulation International.

[19] Z. Lu, Y. Li, J. Wu, S. Zhang, and Y. Zhong. Soams: A novel soa-based system and network management model and scheme. In *Proceedings of the 2008 IEEE International Conference on Services Computing - Volume 2*, SCC '08, pages 479–480, Washington, DC, USA, 2008. IEEE Computer Society.

[20] Z. Lu, J. Wu, and W. Fu. Apply ws-management to manage real resources: Mape use case study. In *Proceedings of the 2011 IEEE International Conference on Web Services*, ICWS '11, pages 696–697, Washington, DC, USA, 2011. IEEE Computer Society.

[21] Z. Lv, J. Wang, Y. Wu, J. Wu, and Y. Zhong. Mws-mcs: A novel multi-agent-assisted and ws-management-based composite service management scheme. In *ICWS*, pages 1041–1042. IEEE, 2009.

[22] Z. Lv, J. Wu, S. Zhang, and Y. Zhong. Research on ws-management-based system and network resource management middleware model. In *ICWS*, pages 1051–1053, 2009.

[23] S. Menzier. Extensible hardware management using ws-management and ipmi. WinHec 2005, 2005.2.

[24] G. Moura, G. Silvestrin, R. Sanchez, L. Gaspary, and L. Granville. On the performance of web services management standards - an evaluation of muws and ws-management for network management. pages 459 – 468, 21 2007-yearly 25 2007.

[25] R. Neisse, R. Vianna, L. Granville, M. Almeida, and L. Tarouco. Implementation and bandwidth consumption evaluation of snmp to web services gateways. volume 1, pages 715 –728 Vol.1, april 2004.

[26] B. T. R. Savarimuthu, M. Purvis, M. Purvis, and S. Cranefield. Integrating web services with agent based workflow management system (wfms). In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 471–474, Washington, DC, USA, 2005. IEEE Computer Society.

[27] Y. Wu, J. Wu, and Z. Lv. System management using web service tech whitepaper. *CHINA Electronics Standardization Institute*, 2008.

[28] S. Z. Yu Wu. A novel wsdm-based soa management model. *International Conference on Next Era Information Networking*, 2007.

[29] Y. Yuan, Y. Wu, L. Zhong, L. Zhihui, and etc. Research report of china soa architecture, public document. *CHINA Electronics Standardization Institute*, Sept. 2009.