This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Evaluating the Performance of Web Services Composition for Network Management

Ricardo Lemos Vianna, Maria Janilce Bosquiroli Almeida,
Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville
Institute of Informatics - Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brazil
Email: {rvianna, janilce, liane, granville}@inf.ufrgs.br

Abstract— **The composition of network management information is a feature widely required but not properly supported in traditional management technologies. In the last years, Web services technology has been investigated to enable more sophisticated management solutions. In this paper, we show that Web services have more to offer to the network management discipline than just bridging established management protocols and Web-based applications. We explore the possibility of using Web services composition for network management considering two approaches: in the first one a single device needs to be contacted and its information composed; in the second one, many devices need to be contacted and the information retrieved from them need to be composed. We show that using proper tools one can not only really use Web services composition for network management, but also that such use can be integrated with traditional management technologies that are unlike to be abandoned in short and mid terms. Moreover, we investigate the performance of Web services compositions for network management considering response time and network traffic. Performance investigations are crucial because Web services protocols are based on plain text XML documents and impose a processing overhead, which may prevent their adoption depending on the requirements and limitations of the management environment.**

I. Introduction

The network management research community has been investigating and proposing management protocols for years. The Simple Network Management Protocol (SNMP) [1] is certainly the most successful one, being widely deployed in today's networks and commonly referenced as the *de facto* Internet management solution. Despite its wide adoption, SNMP has well recognized drawbacks (e.g., related to security, performance, flexibility) that prevent its actual use beyond network monitoring and inventory. Several solutions to SNMP problems have been proposed, but the main issues effectively remain unsolved, at least from the point of view of the majority of network operators.

Recently, Web services technologies have captured the attention of both academia and industry as a potential management alternative to the old and limited SNMP. As a result, several investigations have been carried out in order to understand the impact of using Web services for network management, for example, in terms of bandwidth consumption [2], message delivery delay [3], or ease of use [4]. In industry, major players have been recently standardizing Web services for network management through two main efforts: the WS-Management [5] specification from DMTF (Distributed Management Task Force); and MUWS (Management Using Web Services) [6] from OASIS (Organization for the Advancement of Structured Information Standards). In essence, both efforts define Web services operations used to manage end-systems. Although the main target is Internet servers and user hosts, these solutions can be used to manage core network devices as well, such as routers, switches, NAT boxes, and firewalls.

The motivation for using Web services and its ubiquitous Simple Object Access Protocol (SOAP) [7] in the network management area is that these technologies in fact address problems that SNMP investigations have been trying to solve in years. For example, Web services are far more flexible and the set of available tools allows the fast development and deployment of Web services-based systems. SNMP, on the other hand, requires much more skilled developers with deep knowledge about the protocol specificities, which makes the deployment and broad use of SNMP much more limited.

Another important drawback of SNMP, which is central to this paper, is its lack of proper service composition support. Some attempts to address this issue have been done in the past, but with little success and, most importantly, not being actually adopted by network operators. Service composition is critical to network management because it enables the dynamic definition and deployment of sophisticated management services constructed over a set of more basic services.

This paper extends a previous work [8], where we investigated how network management could have benefits from Web services composition. Now, we are interested in evaluating how those Web services compositions perform, in terms of response time and network traffic. Our approach is to build Web services over traditional network management services and gradually introduce sophisticated services created using Web services composition solutions. We do not want our approach to be decoupled from the wide SNMP base. Thus, our Web services must be able to access target devices using SNMP as the "last mile" management interface. Above basic management Web services we then find more sophisticated ones, defined using the Web Services Business Process Execution Language (WS-BPEL) [9]. With such approach, our main goal with this work is to show that Web services technologies can not only co-exist with SNMP but it can also bring more powerful tools (in this case, composition) to the current network management approaches. Our performance evaluation

1-4244-0353-7/07/$25.00 ©2007 IEEE

is generic and intends to verify the behavior of WS-BPEL compositions for network management when the number of partner Web services and managed devices increases.

In another work [10], we have evaluated compositions in a specific study-case of BGP routers management. There, the same management tasks were performed using three different solutions (WS-BPEL-based compositions, Java *ad hoc* compositions, and the IETF Script MIB) and their performance results were compared in terms of performance and ease of use. We call *ad hoc* compositions when the new composed Web service is constructed using directly a programming language, not using a Web services composition standard.

The remainder of this paper is organized as follows. Section 2 reviews Web services-based management issues, and Web services composition strategies. Section 3 introduces our approach on Web services composition in the context of network management, while Section 4 presents information about our system prototype implementation. Performance issues are addressed in Section 5. Finally, this paper is closed in Section 6, presenting concluding remarks and future work.

II. Background

In this section we first review a technique to bridge SNMP and Web services technologies, the use of Web services to SNMP gateways. Above these gateways one can build additional services then using composition, which is also reviewed.

A. Bridging Web services and SNMP

Although SNMP has its limitations, as discussed before, it is not proper to figure out that Web services will fully replace SNMP. In fact, SNMP-enabled devices will probably remain in networks for years, because replacing SNMP requires device firmware updates or even total substitution, which is expensive and rarely accepted by network operators without complains. The deployment and use of Web services to SNMP gateways, on the other hand, is an intermediate and more feasible approach where SNMP-enabled devices are integrated into Web services-based management systems. In this case, the target devices still use SNMP and do not need to be updated or replaced, while management systems can evolve incorporating the Web services technology (Figure 1).

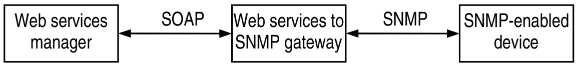

Fig. 1. General view of a Web services to SNMP gateway

The creation of Web services to SNMP gateways can be accomplished by different strategies that define how SNMP operations and management information from MIBs (Management Information Bases) are mapped to Web services operations. These main gateway approaches are:

- **Protocol-level gateway**: SNMP operations are directly mapped to Web services operations [11]. For example, the SNMP *Set* operation is mapped to a corresponding *Set* Web services operation;
- **Object-level gateway**: SNMP management information is mapped to Web services operations [2]. For example, the SNMP *sysUpTime* object usually present in devices' MIB and that holds the amount of time the SNMP agent is running, is mapped to the *GetSysUpTime* Web services operation;
- **Service-level gateway**: SNMP management services are mapped to Web services operations [4]. For example, the set of management objects responsible to control the transferring and execution of management scripts, for example using the IETF Script MIB [12], is mapped to a single *DownloadAndRun* Web services operation.

Protocol-level and object-level gateways are easily built because their code is automatically generated from SNMP well defined elements: SNMP operations in the case of the protocol-level gateway, and SNMP MIBs in the case of the object-level gateway. In order to code a service-level gateway, however, human interpretation is required because the management services exposed by an SNMP-enabled device are not formally defined. This leads to a situation where the same management service can originate different service-level gateways if they are coded by different developers.

In terms of performance, protocol-level gateways impose high bandwidth consumption, while object-level and service-level gateways decrease the bandwidth consumption by aggregating, inside the gateways, SNMP information retrieved from managed devices. Bandwidth consumption can also vary if message compression and encryption are employed. Although service-level gateways are more difficult to be built, they perform better than protocol-level and object-level gateways because they can aggregate more information and deliver them to the Web services-based manager using fewer messages.

B. Web services composition

To advance beyond the Web services basic framework (publishing, searching, and invoking), models for Web services composition are required [13]. Through service composition it is possible to build new services with simpler services already built (contributing also for software reuse). There are two main models to perform Web services composition: orchestration and choreography.

In the orchestration model, there is a main process that coordinates the interaction with other Web services. Such process is a set of activities, or procedures, required to realize a task in an organized order. Orchestration describes how Web services can interact at message level, defining the execution order of the interactions. These interactions can cross the organization boundaries and result in a long-term transactional process. With orchestration, the process is controlled under the perspective of one of the parts involved. The specification WS-BPEL (Web Services Business Process Execution Language) [9], currently under OASIS's arms, appears in evidence among several proposals. WS-BPEL is a specification that models Web services behaviors through an XML grammar that describes the logic needed to coordinate services that participate in a process flow of a task. That grammar is then interpreted and executed by an orchestration engine that coordinates the many process activities and uses a compensation strategy when errors occur. WS-BPEL is the most mentioned specification involving Web services composition and is currently in the version 2.0. Until version 1.1, WS-BPEL was called Business Process Execution Language for Web Services (BPEL4WS).

III. Composing Web Services for Management

In this section we discuss how the Web services to SNMP gateways previously presented can be composed in order to build new, more sophisticated network management services. In our approach, the management Web services composition is realized using the orchestration approach. Although choreography could be used as well, orchestration, and its WS-BPEL specification, is more proper for network management because the network management discipline itself is more hierarchy-oriented, where top-level managers order low-level manager to execute management tasks. These low-level managers may further delegate management tasks to another level of managers below them. Since we have here one management process at one level controlling the execution of other processes in the level below, this behavior can match the orchestration approach more easily than the choreography approach.

A. Automating the Use of WS-BPEL for Network Management

WS-BPEL is a popular composition language that supports the orchestration approach. The services to be composed and the new services resulting from the composition itself must be properly described in WSDL (Web Service Description Language) documents. Since the Web services to SNMP gateways form the mean to access final managed devices, these are the first Web services to be compliant with this available WSDL requirement. Although service-level gateways, as presented before, are more suitable in most of the cases, their generation depends on the human interpretation of the SNMP services and thus cannot be automated. For this reason, we avoid service-level gateways in this work, and prioritize the investigation considering the object-level gateways, that have intermediate performance but whose creation are automated. To comply with the WSDL requirement, a building process developed in a previous work [2] has been extended.

Object-level gateways are created by a *Web-based building system* that receives as input an SNMP MIB description file. The building system reads the MIB file, defined following the SMI (Structure of Management Information) rules, and generates a new gateway and an associated WSDL file, which is then used in the composition process. In our prototype system, gateways are PHP scripts which run in an Apache Web server and use NuSOAP as SOAP library.

Compositions, like service-level gateways, depend on the human interventions to be defined. In order to automate as much as possible the definition of network management compositions, we have coded an additional Web-based building system to help the user that wants to define new management services built upon object-level gateways. Our building tool can create WS-BPEL-based compositions for network management following two approaches: device information aggregation (that retrieves information from just one device), and network information aggregation (that composes information retrieved from different devices located across the same managed network).

B. Device Information Aggregation

In the device information aggregation approach (Figure 2), all composed information is retrieved from the same managed device, even if such information are retrieved using many intermediate object-level gateways. This approach aims at composing information related to different aspects of the same device, and enabling this new composed information to be available to remote Web services managers through a single Web services call. For example, without such composition a Web service manager interested in retrieving a device's description and its list of network interfaces needs to call two object-level gateways, one responsible for descriptive information and the other responsible for network interfaces information. With direct SNMP messages the situation gets even worst, because much more SNMP operations have to be issued. Using this approach, however, all information is retrieved calling a single Web service operation, that one that in fact composes the device's information. Since we assume that gateways and Web services compositions will be placed closer to network devices, it is possible to save bandwidth and decrease the delivery time because the most intensive SNMP communications will be confined near to the device's neighborhood. In this scenario, only two messages will be exchanged between the Web services manager and the new composed Web service. Even if Web services (composed and gateways) and the target device are not close to each other, device information aggregation is still useful, offering an ease-to-use way to access the final management information.

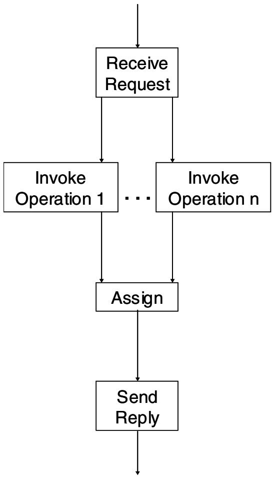

Fig. 2. Device information aggregation

C. Network Information Aggregation

The network information aggregation approach (Figure 3), on the other hand, supposes that the management information required by the Web services manager is distributed across many different network devices. This composition approach has been conceived to enable what is called in the management field "network-oriented management". Network information aggregation approach thus works over a whole network, instead of over a single device, as in the previous approach.

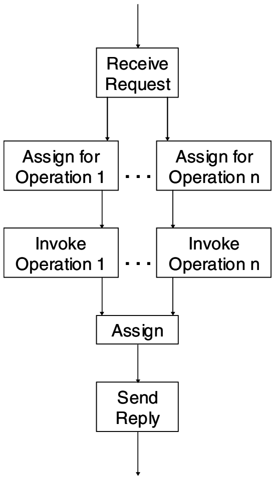

Fig. 3. Network information aggregation

In this network information approach, network managers collect information from many devices exchanging only two messages (a request and a reply) with the Web services composition. That avoids the manager to contact, one by one, each device or gateway involved in the composition, contributing to save bandwidth and decreasing the delivery time, besides simplifying the access to the management information. Network information aggregators need no input parameters. The information required by each operation invoked in the composition (e.g., target device) is statically placed in the BPEL document. For this purpose, a BPEL Assign activity for each operation is created and executed before the operation invocation. To illustrate this approach, we can imagine that the network manager wants to get the list of operating system in use in each device. Instead of calling the Web services gateway operation responsible to return the system description for each device, the manager can use a BPEL composition to perform such a task. As in the previous approach, the results of the invoked operations are collected and merged in a single reply message using a BPEL Assign activity.

IV. Implementation and Software Architecture

A useful composed Web service, of course, has to be ready to run in order to accomplish its task. WS-BPEL-based compositions require a proper software environment in order to offer the new services defined. Since we are considering Web services supported by SOAP layered on top of HTTP, an HTTP services is required to deal with the HTTP messages that carry SOAP contents. Then, Apache Tomcat is used for this purpose. After receiving a request, the HTTP server needs a module able to handle to SOAP contents. Since Tomcat is not able itself to deal with SOAP messages, Apache Axis is required to take care of this required procedure. Finally, each composition requires an execution engine able to implement the new services defined. We are then using the open source ActiveBPEL for this role. ActiveBPEL requires Axis to receive external invocations for services defined in the compositions, and uses Axis when the operations of object-level gateways need to be accessed according to the composition specification. Figure 4 depicts the general environment for composition executions.

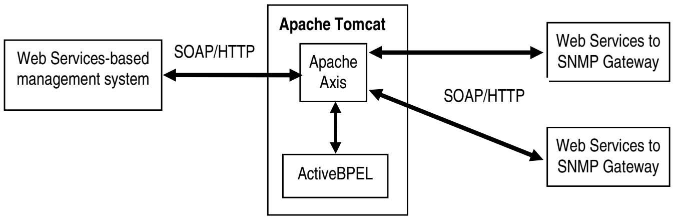
Web Services-based management system
SOAP/HTTP
Apache Tomcat
Apache Axis
ActiveBPEL
SOAP/HTTP
Web Services to SNMP Gateway
Web Services to SNMP Gateway

Fig. 4. Deployment scenario

V. Evaluating Aggregators Performance

Our information aggregation approaches need to be evaluated in order to know whether they can be in fact used for network management. Then, we have evaluated the performance of BPEL compositions focusing in two main aspects: mean response time and network traffic.

Response time was observed in three different points. Application response time has been computed by subtracting timestamps after and before the Web services-based manager invokes the aggregator. That includes the SOAP API and TCP/IP stack overheads. Aggregator response time represents the time between the SOAP request sent by the manager and the associated reply issued by the aggregator. Gateway response time, on the other hand, means the amount of time needed to contact all the partner gateways involved in the composition. Response time of both aggregator and gateways includes the overhead of establishing and finalizing TCP connections. Each measurement has been repeated 30 times to calculate the mean response time, because after this amount of samples we can consider that the variable average has a normal distribution. Network traffic was evaluated in two points: between the manager and the aggregator and between the aggregator and gateways. The second traffic represents the total amount of packages exchanged between the aggregator and all partner gateways. Both traffics include the overhead of manipulating TCP connections as well.

The tests have been performed in a high-performance computing cluster whose nodes are connected through a 100Mbps switch and have the same hardware and software configurations shown in Table I. To simplify the test environment and result analysis, we always used the same gateway in the compositions. This object-level gateway has only one operation, *GetSysDescr* that returns the value of *sysDescr* SNMP object from the MIB-II.

In a first scenario, we are interested in verifying the behavior of device information aggregators when more gateways are

TABLE I

CLUSTER NODES CONFIGURATION

Processor	Pentium III 1.2GHz
RAM memory	1GB
Operating system	GNU/Linux Gentoo (Kernel 2.6.14)
Web server	Apache (2.0.54)
PHP	4.4.0
SNMP	NET-SNMP (5.2.1.2)
Java	J2SDK (1.4.2)
Apache Tomcat	5.0.28
Apache Axis	1.2RC2
ActiveBPEL	1.1
NuSOAP	0.6.3

added to the composition. The number of gateways varied from 1 to 10. The results obtained are shown in Figure 5. In a second scenario, we are interested in verifying the behavior of network information aggregators when more gateways and managed devices are added to the composition. The number of gateway-device pairs varied from 1 to 10. Figure 6 shows the results obtained for this scenario. In both scenarios, each component - manager, aggregator, gateways, and devices - was placed in a different cluster node.

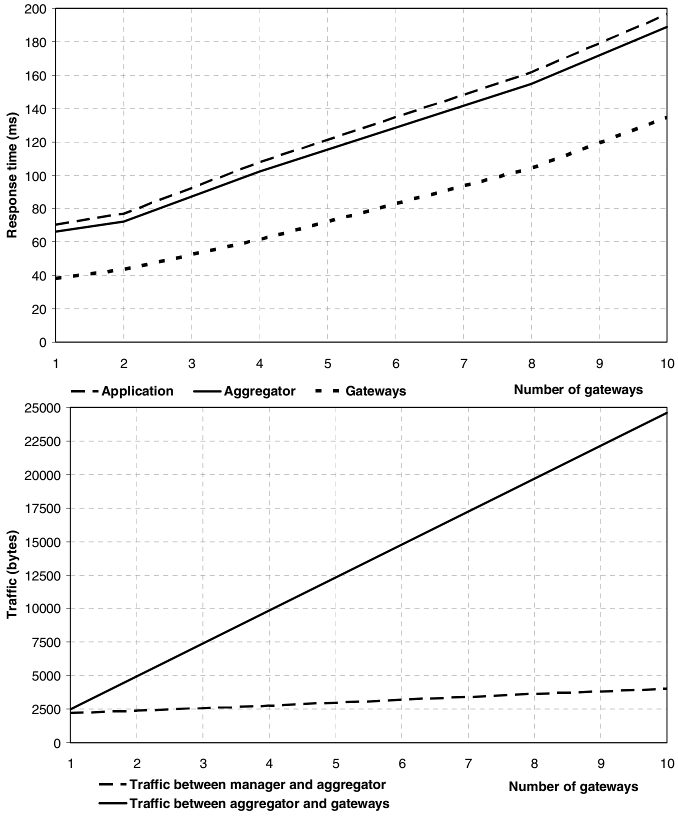

Fig. 5. Device information aggregation performance

Both graphics shows the same behavior considering the increasing number of gateways. The three lines in each graphic grow at different rates, however. It means the overhead (the difference between two lines) increases - even slowly - when more gateways are added to the composition. Network information aggregator had a slightly worse response time because the structure of its BPEL documents is more complex (one more Assign activity to each gateway added is needed) and more managed devices are contacted (avoiding the influence of NET-SNMP caching feature when the same object is retrieved several times). We can note that the processing overhead in the management station - application response time minus aggregator response time - is small and grows slowly with the increasing number of gateways. The overhead introduced by BPEL use - aggregator response time minus gateways response time - on the other hand, is considerable and may be a problem if the management application has real time requirements.

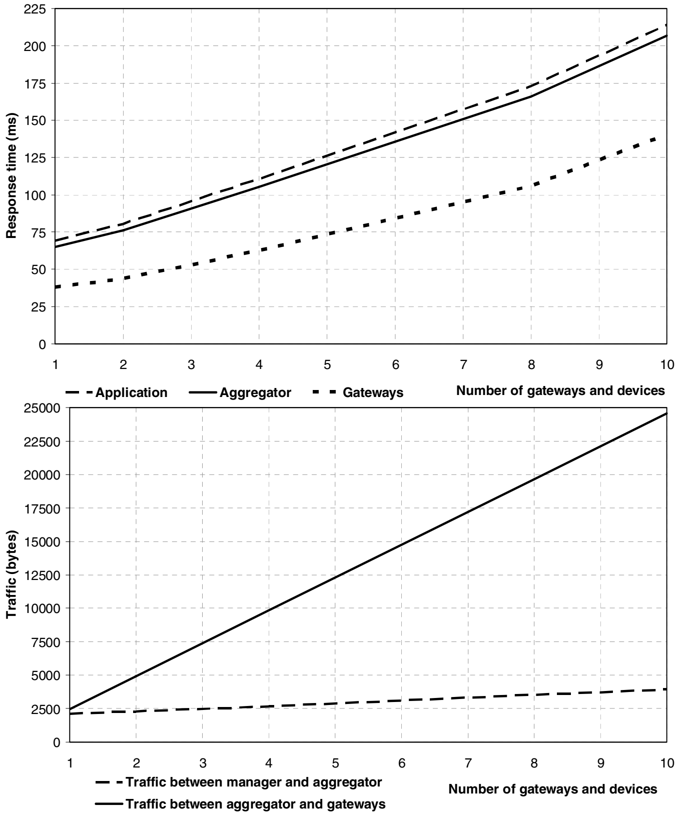

Fig. 6. Network information aggregation performance

Regarding to the network traffic, one can observe that using information aggregators the traffic near the management interface is reduced, if compared with the traffic that would be required in a Web services-based manager directly contacting the gateways. The bandwidth saved will be higher when more gateways are used, because only two messages are exchanged between the manager and the aggregator - SOAP request and reply, instead of exchanging n requests and replies - where n means the number of gateways. Our previous work [2] has shown that the traffic near the management interface could be much higher if an SNMP-based manager is used, depending on the amount of SNMP objects being retrieved. As expected, in both aggregators the traffic exchanged between the aggregator and the gateways is the same, because the only difference between those aggregators is that in the device information aggregator all the gateways contact the same device, while in the network information aggregator each gateway contacts a different device (but retrieving the same information - *sysDescr*). On the other hand, the traffic between the manager and the aggregator is a slightly higher (less than 100 bytes) in the device approach case, because the manager needs to send parameters (device IP and SNMP community string) to the aggregator, which is unnecessary in the network approach (devices' information is statically placed in the BPEL code).

VI. Final Remarks and Future Work

In this paper we have approached the Web services composition from the network management point of view. The main composition standard (i.e., WS-BPEL) has been reviewed. In order to use composition in network management, we proposed two approaches: device and network information aggregation. The first one is suitable when retrieving information from a single managed device. The second approach, on the other hand, can merge management information that is spread across several devices in a network. In addition, we have presented the implementation of a Web-based building system to help network operators that want to define new management services using functionalities offered by object-level gateways. WS-BPEL has been used for this propose. In addition, our approaches for service aggregation have been evaluated considering two issues: response time and network traffic. The tests have been carried out gradually adding more object-level gateways to the composition environment.

Both approaches turn the access to management services much easier, because managers can get all the required information with just one Web services call. Other scenario that can have benefits from WS-BPEL-based aggregation is that one where one needs to retrieve information from devices spread in different administrative domains. Directly using SNMP is difficult in the case, because SNMP traffic is normally confined to the intranet environment and blocked by Internet border firewalls. Then, placing a Web services to SNMP gateway close to each device we can have an aggregator close to the Web services manager that will contact each gateway using SOAP over HTTP/HTTPS - protocols commonly allowed by firewalls, making possible such aggregation.

Regarding the network traffic, other advantage of using Web services composition is that it reduces the network bandwidth consumption near to the management interface once the manager always needs exchanging only two messages with the aggregator, independent of the amount of information aggregated. The heavy interactions between gateways and devices and between aggregators and gateways can be confined near devices neighborhood. The results presented in Section 5 showed that the bandwidth saved will be as higher as greater the number of partner gateways is. In other words, if the amount of information aggregated increases, the bandwidth saved increases too. In terms of response time, the results indicated that there is a considerable overhead introduced by the aggregation process. Although slowly, this overhead grows when more gateways are added to the composition, in opposite to what happened with the network traffic. Thus, real time requirements may prevent the adoption of this solution, depending on the amount of information to be aggregated.

This paper is our third contribution in applying Web services towards real network management service composition. Our next steps include expanding our tool to take benefits from other WS-BPEL structured activities, such as While and Switch clauses. In addition, we are also investigation the use of WS-BPEL fault handler schemes to have more robust composed Web services. Finally, we are planning to include a UDDI registry in our solution, where Web services gateways will be supposed to be registered. With the integration of UDDI support in our system prototype it will be possible, for the users, to query and discover services to be composed.

References

[1] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," Internet Engineering Task Force (IETF), RFC 3411, STD 62, Dec. 2002.

[2] R. Neisse, R. L. Vianna, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco, "Implementation and Bandwidth Consumption Evaluation of SNMP to Web Sevices Gateways," in *NOMS 2004 - 9th IEEE/IFIP Network Operations and Management Symposium*, Seoul, South Korea, Apr. 2004, pp. 715–728.

[3] A. Pras, T. Drevers, R. v.d. Meent, and D. Quartel, "Comparing the Performance of SNMP and Web Services-Based Management," *IEEE eTNSM - eTransactions on Network and Service Management*, vol. 1, no. 2, p. 11, Dec. 2004.

[4] T. Fioreze, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco, "Comparing Web Services with SNMP in a Management by Delegation Environment," in *IM 2005 - 9th IFIP/IEEE International Symposium on Integrated Network Management*, Nice, France, May 2005, pp. 601–614.

[5] A. Arora, J. Cohen, J. Davis, E. Golovinsky, J. He, D. Hines, R. McCollum, M. Milenkovic, P. Montgomery, J. Schlimmer, E. Suen, and V. Tewari, "Web Service for Management (WS-Management)," Distributed Management Task Force (DMTF), Apr. 2006, http://www.dmtf.org/standards/published_documents/DSP0226.pdf.

[6] OASIS, "Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1," December 2004, http://docs.oasis-open.org/wsdm/2004/12/muws/cd-wsdm-muws-part1-1.0.pdf.

[7] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web Services Architecture," Feb. 2004, w3C Working Group Note 11 - http://www.w3.org/TR/ws-arch.

[8] R. L. Vianna, M. J. B. Almeida, L. M. R. Tarouco, and L. Z. Granville, "Investigating Web Services Composition Applied to Network Management," in *ICWS 2006 - IEEE International Conference on Web Services*, Chicago, Illinois, USA, 2006, pp. 531–537.

[9] A. Alves, A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, C. K. Liu, D. Knig, V. Mehta, S. Thatte, D. van der Rijn, P. Yendluri, and A. Yiu, "Web services business process execution language version 2.0," May 2006, http://www.oasis-open.org/committees/download.php/18714/wsbpel-specification-draft-May17.htm.

[10] R. L. Vianna, E. R. Polina, C. C. Marquezan, L. Bertholdo, L. M. R. Tarouco, M. J. B. Almeida, and L. Z. Granville, "An Evaluation of Service Composition Technologies Applied to Network Management," in *IM 2007 - 10th IFIP/IEEE International Symposium on Integrated Network Management (to appear)*, Munich, Germany, May 2007.

[11] J. Schoenwaelder, A. Pras, and J. P. Martin-Flatin, "On the Future of Internet Management Technologies," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 90–97, Oct. 2003.

[12] J. Schoenwaelder, J. Quittek, and C. Kappler, "Building Distributed Management Applications with the IETF Script MIB," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 5, pp. 702–714, May 2000.

[13] F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana, "The Next Step in Web Services," *Communications of the ACM*, vol. 46, no. 10, pp. 29–34, 2003.