Interaction Translation Methods for XML/SNMP Gateway*

Yoon-Jung Oh[1], Hong-Taek Ju[2], Mi-Jung Choi[1], and James Won-Ki Hong[1]

[1] Dept. of Computer Science and Engineering, POSTECH, Korea
[2] Dept. of Computer Engineering, Keimyung University, Korea
{bheart, juht, mjchoi, jwkhong}@postech.ac.kr

Abstract. XML-based network management has been proposed as an alternative or to complement SNMP-based network management. But the XML-based network management does not yet provide a method to manage networks equipped with legacy SNMP agents in the integrated management system. This integrated management system must include an XML/SNMP gateway, which translates and relays messages between the XML-based manager and the SNMP agent. In this paper, we propose three methods for interaction translation in the gateway. First, we propose a DOM-based translation, which provides a method for XML-based manager to directly access management information through the standard DOM interfaces. Second, we propose an HTTP-based translation using URI extenstion with XPath and XQuery. Finally, we apply SOAP, which is accepted as a standard protocol for XML, and propose a translation method for the gateway to advertise the translation services to the manager using SOAP RPC. We also compare the advantages and disadvantages of the proposed translation methods.

1 Introduction

SNMP [14] is the most widely used method for network management on the Internet. However SNMP is insufficient to manage continuously expanding huge networks because of constraints in scalability and efficiency. An XML-based network management (XNM) [2, 3] has been proposed as an alternative or to complement SNMP-based network management. XNM has many advantages. First, XML [7] provides powerful and extensible modeling features for structured management information, and XNM can transfer management data in an efficient and reliable manner using HTTP [3]. Also, the management processes such as analyzing, storing and presenting management information can be easily implemented using XML-related technologies [4].
As depicted in Figure 1, four possible combinations between managers and agents can be considered [4]. The gateways in the second and third combinations translate messages and operations between different management schemes. The first and last combinations are typical SNMP and XML-based management framework. The third combination is the most possible in real management framework, because

* The authors would like to thank the Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

M. Feridun et al. (Eds.): DSOM 2002, LNCS 2506, pp. 54–65, 2002.

XML-based manager can manage network devices using legacy SNMP agents in the integrated management. In this combination, an XML/SNMP gateway translates and relays messages between an XML-based manager and an SNMP agent. The gateway must provide both specification translation and interaction translation between the two management applications. Recent research on SMI to XML Schema translation [3, 5] provides the foundation for the XML/SNMP Gateway. However, no concrete or standardized interaction translation method for the gateway exists yet. Most XML-based applications implement various communication methods between peer systems because XML does not specify any standard communication protocol. Therefore, research on concrete and verified methods for the XML-based systems to efficiently exchange messages is necessary.

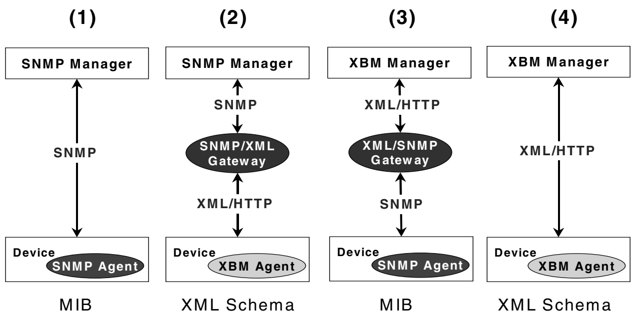

Fig. 1. Interaction Combinations of Managers and Agents

In this paper, we propose three methods for efficient interaction translation in the XML/SNMP Gateway for XML-based integrated network management. These translation methods enable different implementations of XML-based manager to communicate with SNMP agents in a standardized manner. First, we propose a DOM-based translation. The gateway uses the DOM structure and its interfaces to translate management information between XML and SNMP. We analyze the functions and meanings of the DOM interfaces on management information, and translate the interfaces into appropriate SNMP operations. We also propose an HTTP-based translation using URI extenstion with XPath and XQuery, which enables to easily define detailed request message over HTTP between the manager and the gateway. This method improves efficiency in XML/HTTP communication, which is the most common in the exchange of XML documents. Finally, we apply Simple Object Access Protocol (SOAP) [18], which is accepted as a standard protocol for XML, and propose a SOAP-based translation between XML-based manager and XML/SNMP gateway. The gateway advertises its translation services to the manager using SOAP RPC [18]. These three methods are considered to cover overall operation schemes for processing and exchanging XML-encoded information.

The remainder of this paper is organized as follows. In Section 2, we present an overview of XML technologies, and examine related work on SNMP to XML translation. In Section 3, we describe the architecture of the gateway. Section 4 describes the details of proposed methods of interaction translation for the XML/SNMP Gateway. We analyze and compare the methods in Section 5. Finally, we summarize our work and discuss directions for future research in Section 6.

2 Related Work

In this section, we first explain XML technologies, such as DOM, XPath, XQuery, and SOAP, which are applied to our proposed translation methods. We also describe SNMP to XML translation.

2.1 XML Technologies

Extensible Markup Language (XML) [7] is a meta-markup language, which is a W3C standard for document exchange on the Web. Document Object Model (DOM) [8] is a platform- and language-independent interface that allows programs and scripts to dynamically access and update the content, structure, and style of documents. The XML/SNMP Gateway described in Section 3 can use DOM as an interface to access management data and as intermediate storage for data.

XPath [10] is a language for addressing parts of an XML document. XPath has a compact, non-XML syntax to be used within URIs and XML attribute values, and operates on the abstract, logical structure of an XML document. One important type of XPath expression is a location path, which selects a set of nodes relative to the context node, and can recursively contain sub-expressions that are used to filter a set of nodes. XQuery [11], a query language for XML, is designed to be broadly applicable across all types of XML data sources, such as structured and semi-structured documents, relational databases, and object repositories.

Simple Object Access Protocol (SOAP) [18] provides a simple and lightweight protocol for exchanging structured and typed information between peers in a distributed environment using XML. SOAP defines the use of XML and HTTP to access services, objects and servers in a platform-independent manner.

2.2 SNMP to XML Translation

J.P. Martin-Flatin, proposed SNMP MIB to XML translation models, namely model-level mapping and metamodel-level mapping [3]. A model-level mapping is one in which the DTD is specific to a particular SNMP MIB (set of MIB variables). The XML elements and attributes in the DTD bear the same names as the SNMP MIB variables. A metamodel-level mapping is one in which the DTD is generic and identical for all SNMP MIBs.

In our previous work on the XML/SNMP Gateway, we defined an SNMP MIB to XML translation algorithm, and also implemented an automatic SNMP MIB to XML translator using this algorithm [1]. For validation of the algorithm, we had implemented an XML-based SNMP MIB browser using this SNMP MIB to XML Translator.

Frank Strauss presented a library to access SMI MIB information, “libsmi” [5], which translates SNMP MIB to other languages, such as JAVA, C, XML, etc. This library provides a tool for MIB dump into an XML document based on metamodel-level schema mapping.

3 XML/SNMP Gateway

An XML/SNMP gateway provides a method to manage networks equipped with SNMP agents in an XML-based integrated management system. The gateway converts and relays management data between the XML-based manager and the SNMP agent.

Figure 2 illustrates the architecture of an XML/SNMP Gateway [1]. An XML document in the gateway is created by a specification translator, called MIB to XML translator, which translates MIB definition into XML Schema and an XML document. The XML document is used for management information translation.

The Request Handler analyzes the request from the XML-based manager and calls corresponding method in the XML Parser application. The XML Parser retrieves OIDs, identifiers of the target objects, and delivers them to the SNMP Stack. The SNMP Stack then sends an SNMP request message to the SNMP agent, and sends its response from the agent back to the XML Parser. The XML Parser returns newly updated XML content, and the Request Handler delivers the content to the manager. The Trap Receiver receives SNMP trap messages from the agents, and fills out the XML document with trap information. The XML-encoded trap information is sent to the Trap Reporter. The Trap Reporter then sends the notification message to the manager.

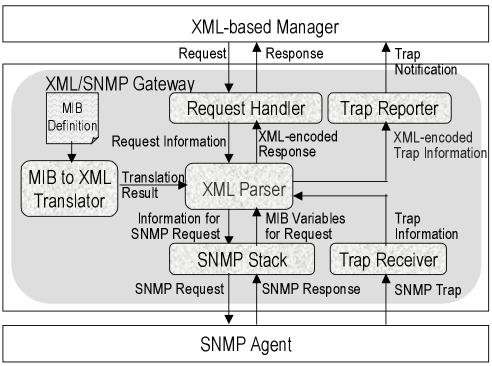

Fig. 2. Architecture of the XML/SNMP Gateway

4 Interaction Translation Methods

In this section, we describe three methods proposed for efficient interaction translation in the XML/SNMP Gateway.

4.1 DOM-Based Translation

The XML/SNMP Gateway transfers management information to the XML-based manager converting SNMP MIB into XML. Using DOM interfaces, the gateway can easily manipulate the structure for information translation. In this method, the DOM tree has a role as an intermediate storage for management data.

In our first approach to interaction translation of the gateway, we propose the DOM-based translation, in which a DOM interface call from the XML-based manager is translated into an SNMP operation. This method is very useful in case of an internal gateway, which is integrated within the management system. The manager can directly access management data in the DOM using DOM API provided by the gateway. In such a case, a key role of the gateway is to maintain DOM contents consistent with the SNMP agent's MIB. In this section, we describe the behavior and the additional meaning of the DOM interfaces in the gateway, and show the translation between DOM interfaces to SNMP operations.

Table 1. Mapping of the DOM Interfaces to SNMP Operations

Interfaces	Methods/Attributes	Translation Results
Document	*GetElementById*()	Access to a node which has a specified value of an attribute "oid" of type "ID".
	GetElementsByTag Name ()	Returns a list of node with a specified node name in the entire MIB module.
Node	*NodeValue*	The value of a node. If the node is a text node and the child of MIB leaf node, send SNMP GET Request to an agent, updates *NodeValue*, and returns it.
	ChildNodes / NextSibling	If the node is a text node and the child of MIB leaf node, send SNMP GETNEXT Request to an agent, updates *NodeValue* of the *ChildNode/ NextSiblings*, and returns them.
Text	*AppendData() InsertData()*	Send SNMP GET request to an agent, appends/inserts a string to the value of a current node, and send SNMP SET with the modified value.

Table 1 shows the mapping of DOM interfaces to SNMP operations based on the XML Schema, which is generated as the result of the specification translation. It includes fundamental interfaces described in the DOM Level 2 core specification [8]. The attributes of the DOM interfaces are mapped to the MIB nodes as the result of a specification translation, and methods are translated into operations, involving SNMP operation. Interfaces excluded in this mapping have no special meaning in this translation method.

As a result of specification translation, a DOM tree in the gateway consists of elements mapped to nodes in a MIB. The value of a leaf node in the MIB is stored in the text node of an element in the DOM. The text node for the value of the leaf node is initialized with a null value. To provide an up-to-date SNMP MIB value from an agent to the manager, it is desirable that the gateway updates the DOM tree whenever the MIB in the agent is modified. However, it is both inefficient and difficult to implement. In our approach, the gateway sends an SNMP request and updates a node value with returned data, only if the manager accesses the text node of a leaf node: that is, the manager invokes *Element::firstChild* or *Element::nextSibling*. In this way, the gateway maintains efficiently the DOM tree consistent with the agent's MIB. In case of a table object, which has variable numbers of instances, an element for the table object in the DOM consists of a list of child text node as the same number of the instances. And these children are created and deleted dynamically as the instances of the table object are changed. Using the SNMP GetBulk operation, we can also improve the consistency of MIB table data. However, the gateway does not eventually solve the inherent consistency problem of the SNMP.

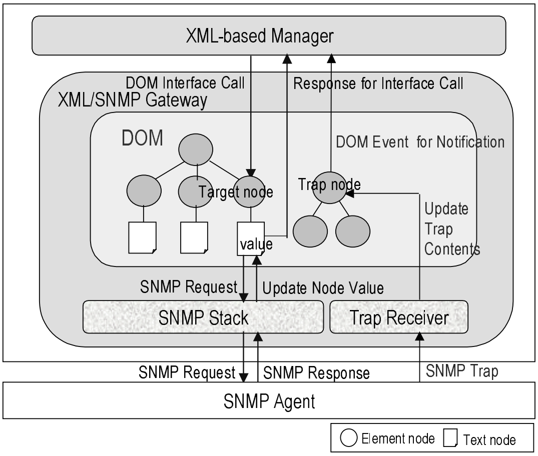

Fig. 3. Interaction between XML-based Manager and Gateway using DOM

Figure 3 illustrates how the manager retrieves a MIB value using DOM API involving an SNMP operation. A DOM interface call from the XML-based manager is translated into an SNMP Request and returns a node value from SNMP response to the manager after modifying the DOM content. For notification delivery, the Trap Receiver in the gateway receives a trap notification from the SNMP agent and updates trap contents in the DOM. The DOM interface invokes the pre-defined event handler after changes in the DOM trap nodes in order to notify modification of the trap content, and delivers it to the manager.

As described in Table 1, the DOM interfaces provide methods to access a node in the DOM tree by its attribute of type “ID” along with an element name. It is easy to traverse the hierarchy without the “oid” operation using the interfaces. The DOM Traversal Interface provides different logical views using a filtering interface, and the DOM Range Interface provides methods to access and manipulate the document tree within a specified range [9]. This translation method can extend its functionality through these additional features of DOM interfaces, and the XML/SNMP Gateway can easily integrate legacy SNMP agents into XML-based management through the translation between DOM interface-based requests to SNMP requests and vice versa.

4.2 HTTP-Based Translation

In this section, we describe an HTTP-based translation method. The HTTP is a generic stateless protocol, which can be used for many applications in a standardized manner through extension of its request methods, error codes and headers [20]. In this method, the XML/SNMP gateway translates URI-based HTTP requests from XML-based manager to SNMP requests. We extend URI [19] with XPath and XQuery. This method provides efficient ways to retrieve MIB objects in XML/HTTP communication, which is the most common for the exchange of XML documents.

When a manager requests specific management information, addressing the managed objects is important in the request message. We apply XPath [10] and XQuery [11] in a URI string to indicate a target object. XPath is a standard for addressing parts of an XML document and provides a rich addressing mechanism for an efficient and effective query on management information. Table 2 shows a request format and examples using URI extension with XPath and XQuery.

Table 2. XPath and XQuery Expression in URI Extension

Request format using URI extension with XPath and Xquery
http://[gateway_address]/[XPath_expression]?agent=[agent_address] &community=[community_string]&version=[snmp_version] &operation=[operation_type]&query=[XQuery_expression]
Example of XPath expression in URI
http://xml-snmp-gateway.org/ **device[@type="server"]**? agent=141.223.82.72&community=public&version=V1&operation=get
Example of XQuery expression as part of search string in URI
<result> { Let $t := input() //ifTable/ifEntry/ ifType[contains(./text(), "ethernet")] RETURN <totalInOutOctets count="{count($t) }"><in> { sum($t/ifInOctets/text()) } </in> <out> { sum($t/ifOutOctets/text()) } </out></ totalInOutOctets> } </result>

Figure 4 illustrates the translation of HTTP requests into SNMP requests on the basis of XPath/XQuery included the URI string. A CGI-like interface called "Request Handler" in the gateway receives and parses an XPath/XQuery expression as input with several arguments in URI and delivers the request to the gateway application. XPath/XQuery Handler in the gateway application analyzes the expression and returns a list of target nodes. The request for the target nodes in the XML document is translated into SNMP request in the same way as in the previous translation method. For notification delivery, HTTP Client in the gateway sends an asynchronous event message from the Trap Receiver to the manager.

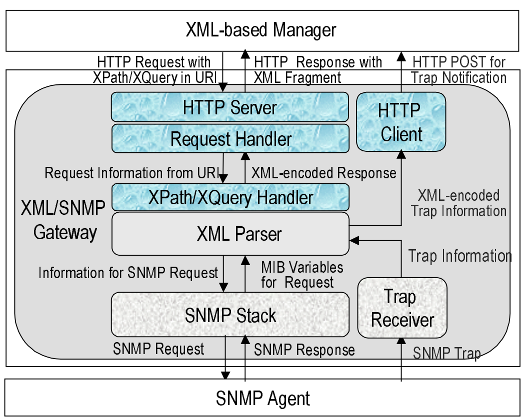

Fig. 4. Interaction Translation of HTTP Request to SNMP Request

Another powerful and structured facility is provided by XQuery, which is a query language that provides query expressions across various types of data sources, such as structured and semi-structured documents, relational databases, and object repositories. XQuery uses XPath as a subset and can easily express a complicated query. For example, when a manager retrieves the total number of in/out octets of a network interface with type “Ethernet” in the interface table, the manager must send a number of requests to retrieve all instances of the objects in the table and then calculate the total number of octets from the returned values. The example in Table 2 simplifies the requests for this case using XQuery. XQuery also provides such features as filtering a document to produce a table of contents, joining across multiple data sources, grouping and aggregating the contents, and querying based on sequential relationships in documents. The XML-based manager can reduce the number of requests and data transfer for further processing in the manager such as complicated statistical analysis. Thus, it can improve management efficiency by adopting XPath and XQuery in a message format.

4.3 SOAP-Based Translation

As we mentioned in Section 2, SOAP [18] is a protocol for exchanging XML-based messages over HTTP or SMTP. At the basic functionality level, SOAP can be used as a simple messaging protocol and also can be extended to an RPC protocol [17]. In this section, we describe a SOAP-based communication as a translation method between XML-based manager and XML/SNMP Gateway.

An XML-based manager exchanges an XML encoded message with communication peers, such as an XML-based agent and an XML/SNMP gateway, which means that a request message from the manager and a response message from the gateway are all formatted as an XML document. SOAP defines a standard method to transfer XML-encoded messages over HTTP.

Table 3. Basic SOAP Messages between XML-based Manager and Gateway

Message	Example
Get Request	<m:getRequest xmlns:m=”http://example.org/xmlsnmp”> <m:community>public</ m:community><m:version>1</m:version> <m:path>// ifSpeed[1]</m:path> </m:getRequest>
Set Request	<m:setRequest xmlns:m=”http://example.org/xmlsnmp”> <m:community>media</m:community>><m:version>1</m:version> <m:path>//hostInfo/hostName</m:path><m:value>zeus</m:value> </m:setRequest>
Response	<m:response xmlns:m=”http://example.org/xmlsnmp” > <rpc:result xmlns:rpc=”http://www.w3.org/2001/12/soap-rpc”> <ifSpeed> 64000 </ifSpeed> </rpc:result><m:response>

We define three types of XML elements for the basic SOAP RPC messages between an XML-based manager and an XML/SNMP Gateway. As described in Table 3, these elements are both named and typed identically to the methods advertised by the gateway and have subelements corresponding to the parameters of the methods. The "getRequest" or "setRequest" has a "version" element indicating the version of SNMP, a "community" to be used for identification, and either an "oid" element for object identification or a "path" element for addressing one or more object nodes in the DOM tree using the XPath expression. The manager can access one or more nodes by specifying the "path" element as well as an "oid". A "query" element is defined to contain XQuery expression for a complicated query. The "setRequest" element uses the "values" element to set a value of a node to be modified. The "response" element is also defined as a response message for "getRequest" and "setRequest". The "response" has the "result" element as the only subelement. The manager finds a method to invoke and to pass appropriate parameters to the method using XML Schema including definitions of these elements.

Table 4. Extended SOAP Messages between XML-based Manager and Gateway

<table>
<tr><th>Message</th><th>Example</th></tr>
<tr><td rowspan="2">GetBulk Request</td><td><m:getBulkRequest xmlns:m="http://example.org/xmlsnmp">
<m:non-repeaters>// ifType[1]</m:non -repeaters>.
<m:max-repeaters>// ifType[1]</m:max -repeaters>.
<m:variables>sysUpTime:ipNetToMediaPhyAddress:ipNetToMediaType
</m:variables></m:getBulkRequest></td></tr>
<tr><td><m:getRequest xmlns:m="http://example.org/xmlsnmp">
<m:path>//sysUpTime | //ipNetToMediaPhyAddress[position() < 3]
| // ipNetToMediaType [position() < 2]</m:path></m:getRequest></td></tr>
<tr><td>Set Scheduled Polling</td><td><m:setSchedule xmlns:m="http://example.org/ gateway">
<m:path>//cpuUsage</m:path><m:initiation>20020101000000</m: initiation>
<m:expiration>20021231240000</m: expiration>
<m:interval>300000</m: interval></m:setSchedule></td></tr>
</table>

As mentioned above, we defined several essential elements, such as “getRequest” and “setRequest”. The SNMP GetBulk operation or other complicated types of request are defined using “getRequest” with XPath or XQuery. Table 4 describes several extended SOAP messages. As described in the previous section, XPath and XQuery provide an efficient way of indicating managed objects to be retrieved. XPath and XQuery can also be applied in the SOAP request message as a parameter of methods.

Figure 5 illustrates a SOAP-based architecture of XML-based manager and the gateway. In this architecture, a SOAP client generates XML encoded SOAP messages, conversely, a SOAP server in the gateway parses the message and invokes an appropriate procedure in the gateway. The SOAP client takes in RPC information from the manager application and generates a SOAP message. The SOAP message is passed to the HTTP Client, which then sends the HTTP POST request to the HTTP Server in the gateway. The HTTP Server in the gateway converts the text POST to the HTTP message and sends the HTTP message to the SOAP server. Then the SOAP server parses the HTTP message into a properly formatted RPC call and invokes the appropriate object served by the gateway. The SOAP server receives the results of the method call and generates a well-formed SOAP response message. The response message backtracks to the SOAP Client in the manager. Finally, the manager application receives the results of the method invocation. For notification delivery, the SOAP Client in the gateway sends an asynchronous event message from the Trap Receiver to the SOAP Server in the manager.

It is very common to exchange XML encoded data over HTTP. SOAP provides a better solution than a proprietary XML/HTTP because SOAP is an open standard with

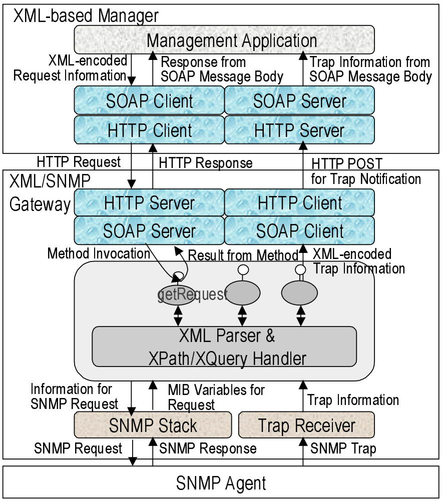

Fig. 5. SOAP-based Architecture of Manager and Gateway

a growing body of developers and vendors supporting it. SOAP defines a standard vocabulary and a structure for messages between communication peers, which eliminates an overhead of parsing and processing messages by a proprietary method in each gateway. XML Schema provides self-documenting features of defining data and interface type like IDL in CORBA and DCOM, and gives the basis of an RPC mechanism. One disadvantage of SOAP is the overhead in converting the native format of application data into an XML-based SOAP message. However, this overhead is eliminated in this approach because the XML/SNMP Gateway already deals with XML documents serialized from DOM tree as a communication message. As more vendors offer SOAP products and services, the advantages of using SOAP will increase.

5 Analysis of the Proposed Methods

In this section, we summarize our research work and discuss the advantages and disadvantages of the proposed methods. Table 5 summarizes them.

In our first approach to interaction translation in XML/SNMP Gateway, we focused on the DOM in the gateway. The gateway uses the DOM structure and its interfaces to translate management information between XML and SNMP. This DOM level translation provides a method to directly access management information through the DOM interfaces for XML-based manager. This method is targeted especially for an internal gateway [6], which is integrated within the manager system. In an internal gateway, the manager's request using the DOM interface is directly translated into an SNMP operation without any request parser application. Instead, the manager is responsible for using DOM interfaces, and thus must understand entire functions of interfaces. The request handler parses a request from the manager and translates it to the DOM interface.

Table 5. Advantages and Disadvantages of the Translation Methods

	Advantages	Disadvantages
DOM-based Translation	No need for request handler between gateway and manager. Can be applied to both internal gateway and external gateway. Uses DOM as intermediate storage for manager.	Imposes a burden on manager of invoking a series of interfaces for request processing in appropriate order.
HTTP- based Translation	Easy to implement using HTTP message extension Provides an efficient mechanism for querying managed objects.	Need of XPath/XQuery parser
SOAP-based Translation	Simple to implement SOAP over HTTP. Inherits advantages in the HTTP-based translation. Provides a standard way to implement RPC.	Overhead of packaging SOAP messages.

The HTTP-based translation method gives an advantage of efficient and effective communication between a manager and a gateway by reducing the amount of request messages, resulting in less data transfer. This approach also can be easily implemented in XML/HTTP with little effort, extending URI with a query expression. To support XPath and XQuery, both the manager and the gateway must include XPath and XQuery enabled module, and which is already supported by commonly used XML parsers [15, 16].

Finally, we proposed SOAP-based communication as a translation method through RPC calls. A structure defined in a SOAP message is translated into a specific method served by the gateway. This method takes in the advantages of the second approach because SOAP is based on HTTP, and also SOAP messages can contain XPath and XQuery expression in its body. Using SOAP, the gateway can receive requests from and send responses to XML-based manager in a standardized way and eliminate the overhead of parsing and processing messages by a proprietary method on XML/HTTP. The manager also has an advantage that it can access remote gateway codes without knowing how to package up XML message and make an HTTP request. Finally, the gateway can communicate with any type of managers, which can understand the interface definition in XML Schema and which can generate a SOAP RPC message from the definition.

6 Conclusion and Future Work

In this paper, we have proposed three interaction translation methods between an XML-based manager and the XML/SNMP Gateway. The gateway uses a DOM tree to process a request from the manager, and to transfer management data corresponding to the MIB variable in an SNMP agent. First, we proposed a DOM-based translation method, which enables the manager to directly access the DOM in the gateway using DOM interfaces in order to exchange management information with the SNMP agent. In the HTTP-based translation, we extended a URI string, which contains information on a request with XPath and XQuery. XPath and XQuery can be easily applied to URIs to express a location path of target objects and to provide a query language in the request message. This method improves efficiency in XML/HTTP communication, which is the most common in the exchange of XML documents. We also proposed a SOAP-based translation method. Using SOAP, the gateway provides a flexible and standardized method for interaction with XML-based manager in a distributed environment. The gateway can communicate with any type of XML-managers, who can understand the interface definition in XML Schema and which can generate a SOAP message from the definition.

We are now in the process of implementing the XML/Gateway and plan to use it for developing an XML-based manager for global element management system. Possible future work includes the performance evaluation of the gateway, and improving the efficiency, performance, and scalability of the gateway.

References

1. J. H. Yoon, H. T. Ju and J. W. Hong, “Development of SNMP-XML Gateway for XML-based Integrated Network Management”, Accepted to appear in the International Journal of Network Management, 2002.
2. H. T. Ju, et al, “An Embedded Web Server Architecture for XML-Based Network Management”, Proc. of the IEEE Network Operations and Management Symposium, Florence, Italy, Apr. 2002, pp. 7–18.
3. J.P. Martin-Flatin, “Web-Based Management of IP Networks and Systems”, Ph.D. Thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), Oct. 2000.
4. M. J. Choi, H. T. Ju and J. W. Hong, “Towards XML and SNMP Integrated Network Management”, Accepted to appear in the 2002 Asia-Pacific Network Operations and Management Symposium, Jeju Korea, Sep. 2002.
5. Frank Strauss, “A Library to Access SMI MIB Information”, http://www.ibr.cs.tu-bs.de/projects/libsmi/.
6. H.G. Hegering, Integrated Network and System Management, Addison-Wesley, 1999.
7. W3C, “Extensible Markup Language (XML) 1.0”, W3C Recommendation, Oct. 2000.
8. W3C, “Document Object Model (DOM) Level 2 Core Specification”, W3C Recommendation, Nov. 2000.
9. W3C, “Document Object Model (DOM) Level 2 Traversal and Ranges Specification”, W3C Recommendation, Nov. 2000.
10. W3C, “XML Path Language (XPath) Version 2.0”, W3C Working Draft, Apr. 2002.
11. W3C, “XQuery 1.0: An XML Query Language”, W3C Working Draft, Apr. 2002.
12. W3C, “XSL Transformations (XSLT) Version 1.0”, W3C Recommendation, Nov. 1999.
13. W3C, “XML Schema”, W3C Recommendation, May 2001.
14. W. Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Third edition, Addison-Wesley, Reading, MA, USA, 1999.
15. Apache XML project, “Xerces Java parser”, http://xml.apache.org/xerces-j/.
16. Gnome project, “The XML C library: libxml”, http://xmlsoft.org/.
17. IBM, “SOAP: Simple Object Access Protocol”, http://www-106.ibm.com/developerworks/library/soap/soapv11.html.
18. W3C, “SOAP Version 1.2”, W3C Working Draft, Dec. 2001.
19. T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax”, IETF RFC 2396, Aug. 1998.
20. DMTF, “Specification for CIM Operations over HTTP”, http://www.dmtf.org/standards/documents/WBEM/DSP200.html