Towards XML and SNMP Integrated Network Management

Mi-Jung Choi*, Hong-Taek Ju and James W. Hong***

*** Dept. of Computer Science and Engineering, POSTECH, Korea**
**** Dept. of Computer Engineering, Keimyung University, Korea**
{mjchoi, juht, jwkhong}@postech.ac.kr

Abstract

SNMP has been the dominating management framework for managing networks and systems on the IP networks. Recently, XML-based network management has been proposed as an alternative or to complement the shortcomings of SNMP-based network management. This paper examine the weaknesses of SNMP-based network management and proposes several approaches to compensate the shortcoming of SNMP using XML technologies. We also present ways to fully take advantages and strengths of using XML technologies wherever possible, especially when managers and agents are being developed for new networks or devices. We also propose the design architectures of XML-based manager and XML-based agent, which can be used in developing XML-based management systems.

Introduction

- Backgrounds
  - Since the late 1980s, the management of IP networks has relied on SNMP because of its advantages: simplicity and interoperability
  - SNMP is inadequate to manage continuously expanding networks because of constraints in both scalability and efficiency
  - Recently, XML-based network management has been proposed for compensating the shortcomings of SNMP-based network management
  - Concrete methods for XML-based network management to compensate or replace SNMP-based network management are lacking
- Goals
  - Propose approaches to compensate the weaknesses of SNMP-based network management using XML technologies
    - Several combinations of SNMP and XML management approaches can exist
    - Examine each approach and propose how XML can be applied to solve the shortcomings of SNMP-based network management
  - Propose the design architectures of XML-based manager and XML-based agent in XML-based management paradigms

APNOMS 2002 - 2 - DP&NM Lab. POSTECH

Since the Simple Network Management Protocol (SNMP) [1] was developed in the late 1980s, the management of IP networks has exclusively relied on SNMP. However, current IP networks are composed of diverse network devices and much more complex. SNMP-based network management has reached its limit to manage the rapidly growing network. That is, as the network expands, the amount of management data to be processed and transferred between managers and agents is continuously increasing. Therefore, SNMP is insufficient to manage these huge and continuously expanding networks because of constraints in both scalability and efficiency. To complement the weaknesses of SNMP-based network management, XML-based network management, which applies Extensible Markup Language (XML) [2] related technologies to network management, is emerging [3, 4]. However, the research in applying the XML technologies to network management is still in early stage. Also, concrete methods for XML-based network management to compensate for and replace the SNMP-based network management have not been sufficiently provided.

In this paper, we first summarize SNMP-based network management and explain the strengths and weaknesses of the SNMP-based network management. We examine the advantages of utilizing XML for network management. Next, to investigate how the advantages of XML can be concretely utilized, we explain the method of applying XML to network management in the manager-agent composition. That is, we divide the method into four approaches: the first approach preserves the existing SNMP manager and SNMP agent, the second preserves the SNMP agent and substitutes an XML-Based Management (XBM) manager for the SNMP manager, the third maintains the SNMP manager and replaces the SNMP agent with an XBM agent, and the last replaces both the SNMP manager and the SNMP agent with both an XBM manager and an XBM agent. Afterwards, we explain the method of applying XML technologies to each approach, and the strengths and weaknesses of the method in detail. Also, we propose detailed design architectures of XBM manager and XBM agent that can be used in XML-based network management.

XML Related Technologies

- XML (Extensible Markup Language): a meta-markup language which was standardized by W3C in 1998 for document exchange on WWW
- XML has two fundamental approaches to define the XML document structure:
  - DTD (Document Type Definition): defines the document structure with a list of legal elements
  - XML Schema: a successor to DTD, supports creation of new data types
- DOM (Document Object Model): a programming interface to access and manipulate XML documents
- SAX (Simple API for XML): event-driven and serial-access mechanism for accessing XML documents
- XSL (Extensible Stylesheet Language): a mark-up language designed for illustrating the method to display XML documents
- XSLT (XSL Transformation): a language for describing how to transform one XML document (represented as a tree) into another
- XPath (XML Path Language): a language for addressing specific parts of an XML document

We briefly explain XML and its related technologies that can be used for network management.

Extensible Markup Language (XML) [2] is a meta-markup language, which was standardized by the World Wide Web Consortium (W3C) in 1998 for document exchange in the World-Wide Web. It is a subset of Standard Generalized Markup Language (SGML) [5]. Therefore, XML has the advantage of easy transmission using HTTP [6] like HTML [7] and has the advantage of extensibility like SGML. XML is a text-based syntax that is readable by both computer and humans. It is flexible and extensible, allowing new tags to be added without breaking an existing document structure. Based on Unicode, XML provides global language support. Also, it offers data portability and reusability across different platforms and devices. With a wide expression power of documents and data, XML has major potential to become the standard interface for management information.

XML has two fundamental approaches to define the XML document structure: Document Type Definition (DTD) and XML Schema [8]. The DTD is used to specify a content model for each element. The content description is part of the element declaration in the DTD and specifies the order and quantity of elements that can be contained within the element being declared. That is, the DTD is used to specify a property for each element and the relationship between the elements.

However, the DTD does not support complex information modeling, so a new modeling mechanism, XML Schema, was proposed. The XML Schema substantially reconstructed and extended the capabilities found in XML DTDs. The XML Schema is based on XML, so is can be parsed and manipulated exactly like the XML documents through the standard API. The XML Schema supports a variety of data types (44 kinds of basic types), while the DTD treats all data as strings or enumerated strings. The XML Schema also allows inheritance relationships between elements and supports namespace integration.

The Document Object Model (DOM) [9] 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 DOM is an API for valid HTML and well-formed XML documents. Simple API for XML (SAX) [10] is the event-driven and serial-access mechanism for accessing XML documents. A DOM parser parses the XML document and creates a DOM tree, keeping the entire tree in memory at the same time.

SAX reads the XML document in sequential order and generates an event for a specific element. Therefore, if the application calls for sequential access to XML documents, SAX can be much faster than other methods without requiring much system overhead. Whenever an event is generated, the method relevant to the event is processed. An XML processor using the SAX does not create a data structure. Instead, while accessing the XML document, it generates events such as the start of an element and the end of an element. Applications can process operations such as gaining the name and attribute of the element by capturing the event. SAX is an interface to an XML parser rather than an API to one of the data structures that can be built using a parser.

Extensible Stylesheet Language (XSL) [11] is a mark-up language designed for illustrating the method to display XML documents. XML documents describe only contents and the structure of the contents. An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary. That is, XSL supports that XML can separate contents from presentation. XSL consists of two parts: a language for transforming XML documents, and an XML vocabulary for specifying formatting semantics.

The style sheet technology that can transform documents is XSL Transformation (XSLT) [12], which is a subset of XSL technology. XSLT fully supports the transformation of an XML document from one format into another, such as HTML or another custom XML document type. The reason for publishing XSLT specification separately from XSL is that XML documents can be displayed, providing an easy display format for end users through transforming the XML documents without formatting semantics.

XML Path Language (XPath) [13] is a non-XML language used to address parts of an XML document. It is a non-XML language because it is not described using XML constructs, such as elements and attributes, even though it is used to address XML documents. The syntax used by XPath is designed for use in URIs and XML attribute values, which means it must be concise. The main premise behind XPath is the traversal of an XML document to arrive at a given node. This traversal is accomplished using expressions. The element can be searched and selected by the address through XPath. Therefore, XPath is not used individually but forms the basis for document addressing in DOM and XSLT.

SNMP-based Network Management

- SNMP-based Network Management
  - Management information model: SNMP SMI, MIB
  - Management communication protocol: SNMP over UDP
  - Management operations: Get, Set, Trap
  - Addressing (Naming conventions): Object IDentifier (OID)
  - Strengths of SNMP-based NM: interoperability, simplicity, stable information meta-model
- Weaknesses of SNMP-based Network Management
  - Scalability and Efficiency
  - Security
  - Poor management information model
  - Difficult to develop management functionality

SNMP-based network management involves initializing, monitoring, and modifying the operation of networks and elements connected to the network using SNMP as the management protocol. The SNMP-based network management has adopted the manager-agent paradigm. The SNMP manager exchanges management data with the SNMP agent through SNMP over UDP. The data transferring methods are polling and trap. Polling is a request/response mechanism, and the trap is an event driven notification. The SNMP supports three basic operations: get, set, and trap. The information model of SNMP, known as Structure of Management Information (SMI) [14, 15], is defined as tree-structured with simple variable types and based on the object-based paradigm. Real management data are defined as Management Information Bases (MIBs) based on SMI. The name convention for MIB variables in tree structure is the OID.

SNMP has many advantages to the extent of being the most popular network management approach in the 1990s. Interoperability is undeniably the main strength of SNMP, and simplicity is the second main achievement of SNMP and the main reason for its success. Another strength of SNMP is that it has a stable information meta-model, SMI. However, as the number of networks and systems becomes extensive, the SNMP-based network management has exposed its drawbacks to manage the immense networks. Its weaknesses were well analyzed in J. P. Martin.-Flatin's work [4].

- **Scalability and Efficiency**: A major problem with SNMP is scalability and efficiency. Scalability refers to the number of agents which can be managed in a single manager system, and efficiency means how quickly and effectively a system performs in such operations as the delivery and processing of data. The problems of scalability and efficiency become an issue in the increase of management data. In the case of less data to manage as the past, the 'simplicity' of SNMP was a great advantage to design and implement a network management system (NMS). However, the network has evolved enormously, and management data exchanged through the network and processed in managers and agents, also increased in tandem. Thus, it is increasingly problematic to manage huge networks with SNMP-based NMS.
- **Security**: Security is an important concern in network management, especially when it involves equipment configuration or administration. It is necessary to limit access to devices to a specific set of users. Simple authentication and access-control mechanisms are the preferred method to provide primary security. SNMPv1 and SNMPv2 support a very simple security mechanism of identification based on a community string. The community string plays the role of a password, and the message from the SNMP manager includes the community string. Security based on a community string is simple. Worse, the messages containing the text-based community string are transferred without encryption, thus they can be exposed to the network easily. SNMPv3 [19] strengthened the security. However, SNMPv3 is not generally used yet. Therefore, SNMP reveals a security weakness.
- **Poor management information model**: As mentioned previously, a merit of SNMP is that it has a stable information meta-model, called SNMP SMI. However, The SNMP SMI is insufficient to present management information because of its limited data type. SMIv1 [14] is still used by MIB-II and most other MIBs now use SMIv2 [15]. However, these two versions are not as complete as the information meta-model. For completeness, a third version of SMI, called SMIng [20], was proposed in 1999. Still confined to the research community, it has not been adopted by the SNMP market. MIBs constitute the main element of the SNMP information model. However, MIBs offer only low-level APIs, which are often called instrumentation MIBs. The main weak points in SNMP are the absence of high-level MIBs, the data-oriented nature of the SNMP information model, and actions to an object programmed by side effect [4]. Therefore, the management information model of SNMP does not sufficiently present the management information and does not act efficiently.
- **Difficult to develop management functionality**: From the viewpoint of developing management functionalities, SNMP is simple. Yet SNMP is not one but several, and there is confusion between the versioning schemes of PDUs, protocols, and management architectures. There is no MIB discovery protocol. It is difficult to maintain the manager's knowledge of an agent's MIB version synchronized with the actual version of this MIB in that agent. In general, as SNMP-based management provides device- and vendor-specific management GUIs to all existing management platforms, it needs to separately develop client programs according to the platform. SNMP expertise is domain specific, hence rare and expensive. Further, there is no specific API for implementing management functionalities. Operations such as insertion, deletion, and searching on a database are basic work in management systems. In SNMP-based management, there are limited supports for third-party DB. The developers of the SNMP manager use a specific DB and the proprietary APIs offered by the DB for best performance. Therefore, standard API for implementing management functionalities, and standard API between the manager and the DB is necessary for saving development cost and a flexible data repository. Therefore, it is both costly and time-consuming, and difficult to develop the SNMP manager [4].

Advantages of XML-based NM (1)

- XML-based Network Management
  - utilizes the advantages and strengths of XML-related technologies and applying them to network management
- Processes of network management:
  - Management information modeling
  - Management information distribution (communication)
  - Storage & analysis of management information
  - Presentation of analysis result
- Advantages of XML-based Network Management
  - Management information modeling: Uses XML DTD or XML Schema for management information model
  - Management information distribution
    - Exchanges management data in the form of XML document using HTTP/TCP
    - Operations: HTTP Get, Post, Push mechanism for notification and scheduled polling

XML-based network management performs network management using XML technologies. The purpose of XML-based network management is to provide alternative or complementary approaches for the current management framework in order to solve the problems of current network management or improve the management environment by taking advantage of XML technologies. Network management involves processes in the following order: modeling management information, communicating with manager and agent, storing gathered data to DB and analyzing the data, and presenting the analysis result to users. Here, the advantages gained from applying XML to each process of network management are examined.

• **Management Information Modeling**: As the XML is becoming ubiquitous, it offers a wide range of skills, as well as tools and a system integration mechanism which its broad applicability provides. And the expressive potential of XML makes it possible to represent a wide range of information models presented in existing management standards and deployed solutions. The XML DTDs and Schemas define the data structure of XML documents. The XML Schemas are devised for overcoming the limit of DTDs which cannot define a new data type. They are also flexible and extensible, allowing new tags to be added without changing the existing document structure [21]. The XML Schemas support the presentation of management information by adding new data types to a variety of data types (44 kinds of basic types), and defining the document structure of management information. Therefore, XML has a major advantage in the management information model through the use of XML Schemas. The existing management information models such as SNMP, CIM, GDMO, can be translated to the XML Schemas, and research on these branches have considerably progressed.

• **Management Information Distribution (Communication)**: For information transmission, XML documents can be easily transferred by HTTP. For compatibility, XML supports the exchange of information between all hardware and software platforms, which support HTTP. That is, the management information of XML document format is distributed through HTTP over TCP. TCP for transport protocol instead of UDP makes it possible to transmit reliable management data and very large application-messages without limitations in message size. Therefore, the problems of using the unreliable transport protocol can be solved. And much management data can be transferred between the manager and the agent, with the whole management value, such as table format. The XML data through HTTP can be compressed by the option of an HTTP header. By compressing data, we significantly reduce network overhead while keeping almost the same latency of end-host for compressing and decompressing.

The operations of HTTP are Get and Post. The HTTP Get operation gathers management information from the agent like the SNMP Get, and the HTTP Post updates the management information. The response of the HTTP Get is XML documents with management information; the response of the HTTP Post includes the 'HTTP OK' message. XPath is used for addressing managed objects. A manager can query effectively about managed objects of an agent using XPath. XPath expressions are formed using element name, attributes, and built-in functions. In a Get operation, a single value can be retrieved by parameters of XPath, or multiple values can be simultaneously retrieved. Also, we can retrieve specific information with conditioning and filtering. The XML-Based Management (XBM) agent sends a notification to the XBM manager by a Push mechanism using a Post operation. J.P Martin-Flatin suggested this push mechanism [4]. For supporting this mechanism, the XBM agent must include a Web client.

Typically, managed data needs to be periodically monitored. The XBM manager sends subscription information to the XBM agent. After receiving the subscription information, the XBM agent schedules a series of message distributions and sends it to the subscriber to the schedule through the Push mechanism using a Post operation. The management data transferred periodically from the XBM agent to the XBM manager can reduce network overhead because it is not based on polling [4]. Also, as XML-based network management performs the basic management functionality through HTTP operation: Get and Post, it is not necessary to develop a new management protocol.

Advantages of XML-based NM (2)

– Storage & analysis of management information
  - o Third-party DB support
  - o Uses standard XML technologies (DOM, XPath, etc.) for processing the analysis of the data
  - o Can perform management operations by manipulating the XML document using the standard DOM interface

– Presentation of analysis result
  - o XML separates the contents of document from the manner of expression
  - o XSLT transforms XML into HTML or another XML document
  - o If the management information is in XML format, Web-MUI is easily inferred from the XML document

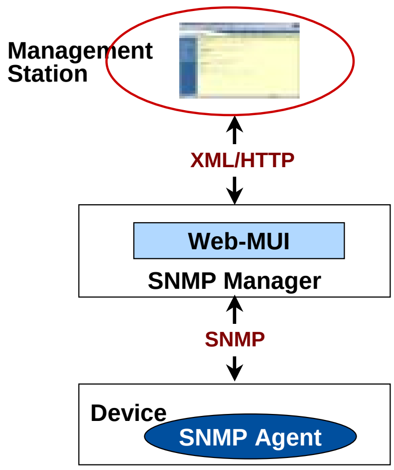
• **Storage and Analysis of Management Information**: XML supports a standard API for accessing XML documents such as DOM, XPath. Using the standard API has many advantages in processing the management data defined in XML documents. A DOM Parser parses the management information and creates a DOM tree. Then the XBM manager can access the chosen node in the DOM tree through the XPath. Through the DOM, we can access any elements of the XML document, and modify the content and the structure of the XML document from the XML input parameters. That is, we can perform management operations such as adding new devices, and setting the configuration of the management server, by manipulating the XML document using the standard DOM interface. It is easy to extract the necessary management data and analyze the data. Therefore, we can analyze the management information using the DOM API.

Because XML data can be easily stored to and accessed from databases (DB) [22] with the support of various available tools, DB processing in network management functionality is easily achieved. Data presented in DOM tree can be easily stored to a DB, and simply transferred to applications. This method reduces the development cost and time, as well as programmers' program overhead. Moreover, there is a lot of freely available software for XML, such as the XML parser, DOM, XPath, Simple API for XML (SAX), etc.

• **Presentation of Management Information**: After the analysis of management information, the next step is to present the analysis result to the user. XML, unlike HTML, separates the contents of document from the manner of expression. A standard exists that XML data validated from XML DTDs or XML Schemas, management information model, are transformed to HTML or another XML document through XSL and XSLT. Using the software supporting the transformation XML to HTML or other display format makes it possible to easily provide Web-based management user interface (Web-MUI). Therefore, if the management information is in XML format, Web-MUI is easily inferred from the XML document, and the XML document can be transformed to another.

The SNMP Manager & Agent Approach

- Composition of SNMP manager and SNMP agent
  - Case: Maintain the current the SNMP manger and the SNMP agent
  - Complementary method: management user interface, namely, presentation part by translation of management information to an XML document
  - Strength: provides a ubiquitous, user-friendly, and cost-effective Web-MUI
  - Weakness: Web-MUI needs to process the translation of SNMP MIB to XML



To analyze carefully the methods of complementing the drawbacks of SNMP-based network management, we divided four approaches of manager and agent: SNMP manager and SNMP agent, XBM manager and SNMP agent, SNMP manager and XBM agent, XBM manager and XBM agent. We describe each approach, and methods to apply appropriate XML technologies to each approach for compensating the weaknesses of SNMP.

The existing SNMP-based network management consists of an SNMP manager and an SNMP agent. The first approach that XML technologies supplement the drawbacks of SNMP is to complement the SNMP manager and SNMP agent unchanged. In this approach, the complementary method is on the side of management user interface, namely, the presentation part.

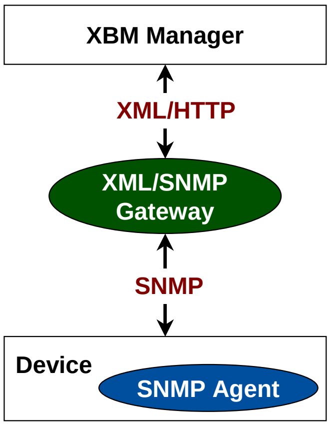
SNMP clarifies the management information model and management protocol as the standard. However, it does not specifically mention a display mechanism of management information. The existing way is to provide a proprietary and a device specific GUI. If a new management system appears, it must design and provide management user interface separately from the MIBs. It is expensive and time-consuming to develop device- and vendor-specific management GUIs for all existing management platforms. Therefore, it is necessary to provide an effective method for supporting a management user interface. For the management user interface, the XML property, which separates the content and the presentation, can be applied. The advantages of XML are used from the viewpoint of the presentation mentioned in Slide 6. The SNMP management information is translated to an XML document, and the XSL transforms the display format from the XML document to HTML, and the HTML provides the Web-based management user interface (Web-MUI).

The figure in the slide shows the complementary method of SNMP manager and agent composition. XML provides a Web-MUI. That means that it supports platform-independent, user-friendly and ubiquitous management user interface through a Web browser and reduces the development effort of a client program. For supporting the Web-MUI, the front-end Web server needs to process the translation of management information to XML document.

As XML provides a Web-MUI, it also supports platform independent, user-friendly and ubiquitous management user interface through a Web browser and reduces the development effort of a client program. For supporting Web-MUI, the Presenter needs to process the translation of SNMP MIB to XML. It is a part of specification translation of XML/SNMP gateway [24]. The research on the gateway is progressed, so can be applied to this approach.

The XBM Manger & SNMP Agent Approach

- Composition of XML-Based Management (XBM) manager and SNMP agent
  - Case: SNMP agent is already embedded and the requests of new NMS happen continuously, the best method is to develop an XBM manager as an alternative of the SNMP manager
  - Complementary method: Uses an XML/SNMP gateway, storage and analysis using the standard API as well as presentation part
  - Strength: solves the scalability problem in the manager's processing capacity, develops easily and quickly management functionalities
  - Weakness: the XML/SNMP gateway is requisite and can be bottleneck, network overhead between the gateway and the SNMP agent



APNOMS 2002 - 8 - DP&NM Lab. POSTECH

An SNMP agent is already embedded and manages most network devices. It is almost impossible to change the agents in already deployed devices. The latest requests on the network management incite developers to develop a new management system without modifying the existing agent. New network devices are continuously being developed, and the corresponding management system is often newly developed. New network devices are basically embedded an SNMP agent. In this case, the best appropriate method is to preserve the SNMP agent and develop an XBM manager as an alternative of the SNMP manager for utilizing the advantages of XML technologies.

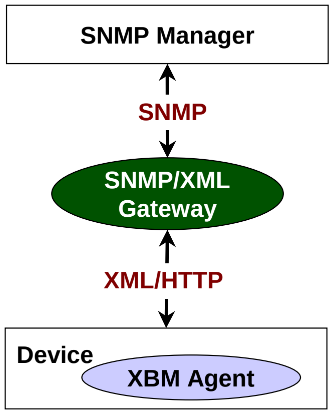
To communicate and exchange the management information between the SNMP agents and the XBM manger, an XML/SNMP gateway is necessary. Research on the gateway is progressing and solutions are being developed [23]. The XBM manager possesses the management information as the XML documents through specification translation of SNMP MIBs to XML Schemas in the gateway. The XBM manager polls and processes the data by using standard APIs such as the DOM interface and the XPath for handling XML document. Moreover, there is a lot of freely available software tools for XML, such as XML parsers, DOM, XPath, SAX, etc. Also, because the XML has the support of third-party databases, it is easy to develop basic data storage and retrieval in implementing the management functionalities. In the presentation, it is possible to extract the management information from the XML documents and provide a Web-MUI. This figure illustrates the architecture of the XBM manager and SNMP agent approach. The communication protocol used between the XBM manager and the gateway is HTTP, and between the gateway and the SNMP agent is SNMP.

From the view of the manager, by replacing the SNMP manager with the XBM manager, the XBM manager solves the scalability problem in the manager's processing capacity because the overhead for processing SNMP is diminished and the number of SNMP messages for transferring management data is also reduced. And the management functionalities can be easily and quickly developed from the support of the standard API and the database. Also, the XBM manager can manage the existing SNMP agents, which are widely used and managed through the XML/SNMP gateway.

However, in this approach, the XML/SNMP gateway is needed, separate from the manager and agent. Because all communications between the XBM manager and the SNMP agents are through the gateway, the gateway must process the translation efficiently, so it must not delay the message transmission between the manager and agent. The communication protocol between the gateway and the SNMP agent is SNMP. This generates the same problems with traditional SNMP-based network management with respect to the communication.

The SNMP Manager & XBM Agent Approach

- Composition of SNMP manager and XBM agent
  - Case: the XBM agent is already developed and the SNMP manager must support the XBM agent
  - Complementary method: need an SNMP/XML gateway, not for compensating but for increasing the interaction
  - Strength: management information using the XML Schemas and management capability in the XBM agent
  - Weakness: the necessity of the SNMP/MXL gateway & translating XML schema to SNMP SMI is difficult, so insufficient research in SNMP/XML gateway



In this approach, the SNMP manager manages the XBM agent. As mentioned before, because it is difficult to change existing agents to another, this architecture is unlikely to be used widely. But this architecture is necessary in the special case that the XBM agent is already developed and the SNMP manager must support the XBM agent. That is, this case is not to compensate for the weaknesses of SNMP, but to increase the interaction between the SNMP manager and the XBM agent. An SNMP/XML gateway is necessary for the SNMP manager to manager the XBM agent, as shown in the slide.

XBM agent defines the management information through the XML Schemas, so it is possible to define various management information model. As parsing the XML document and accessing the managed objects in the XML document are supported in the XBM agent, the XBM agent can process various operations such as filtering and scoping, besides the get/set operations. The XBM agent can share the manager’s processing job and process the management function by itself. So the XBM agent can reduce the manager’s burden. However, an SNMP/XML gateway, which needs the specification translation from the XML Schemas to SNMP SMI and the interaction translation from SNMP operations to XML operations, is required for interacting with the XBM agent with management data defined in XML Schemas. Because the XBM agent is uncommon, so the necessity of the SNMP/XML gateway is rare, research or development for this is not yet reported. Also, the gateway must support translating XML schema to SNMP SMI, but this is very difficult because the XML Schemas are more complex than the SNMP SMI in data types and structures.

The XBM Manager & XBM Agent Approach

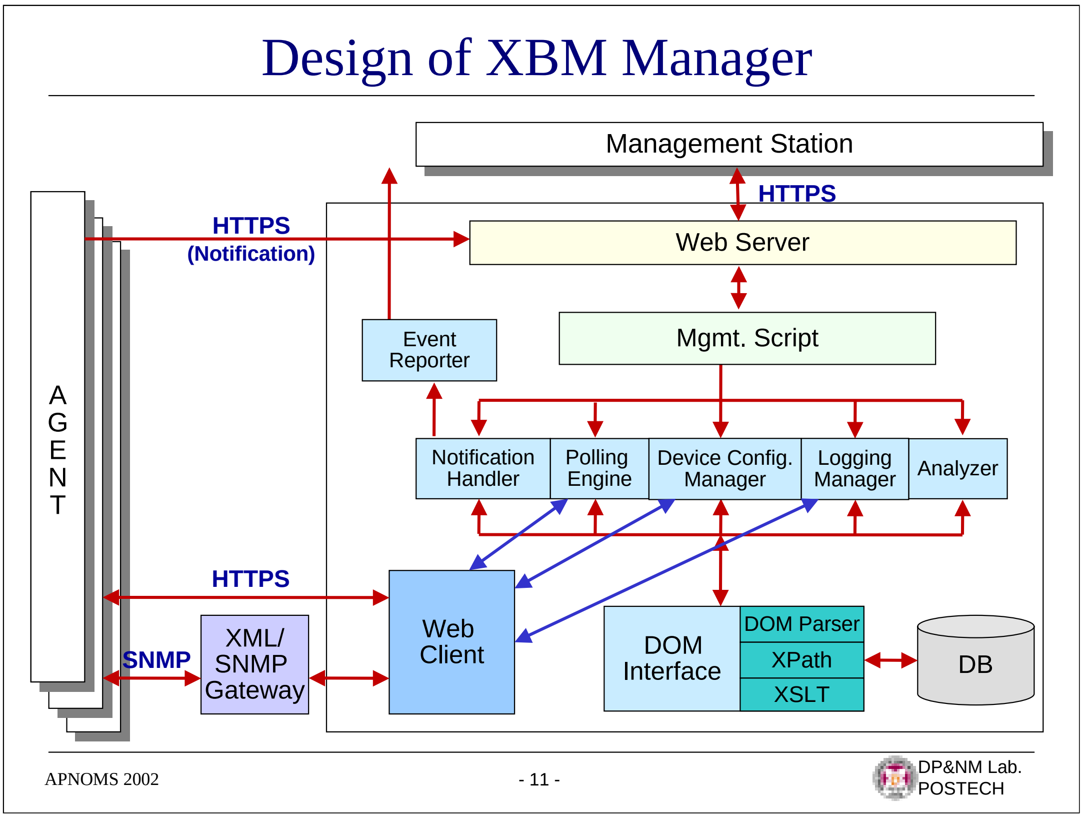
- Composition of XBM manager and XBM agent
  - Case:
    - replaces both the SNMP agent and the SNMP manager with both the XBM agent and the XBM manager
    - develops a new manager and an agent in the closed environment, without considering the existing systems
  - Complementary method:
    - XML technologies can be applied the entire process of network management
      - modeling - XML Schema
      - distribution - HTTP/TCP
      - storage & analysis – DB support & standard API (DOM & XPath)
      - presentation - XSLT
    - Also, complement the non-functional requirements such as security and scalability
  - Strength: provides the maximum advantages of applying XML technologies and solves the drawbacks of SNMP in security and scalability
  - Weakness: This architecture assumes the limited situation, which eliminates the interoperability

This architecture, which replaces the SNMP agent and the SNMP manager with the XBM agent and the XBM manager, is the most ideal case to utilize the maximum advantages of XML-based network management. This architecture, however, can be applied to limited situations, which eliminates the interoperability with the existing SNMP-based network management system. When developing a new manager and an agent in a closed environment, without considering the existing systems, the best method to take advantages of XML technologies applied in the network management is that the XBM manager manages the XBM agent.

This architecture has all the advantages of XML technologies previously mentioned. The advantages are as follows. The supplementing methods of presentation (namely, management user interface) and storage and analysis of management information are the same as those explained in Slide 7 and 8, respectively.

The XML Schema, the management information model of XML, instead of the management information model of SNMP SMI, defines the entire management information that users want and supplement the weakness in presenting management information using SNMP SMI. From the communication viewpoint, because XML uses the HTTP over TCP, it is possible to transmit reliable management data and very large application-messages without the limitation of the message size at one time and to reduce transferred management data by compressing through the option of an HTTP header. This can decrease network overhead. XML-based network management complements weaknesses on the non-functional side, such as scalability, efficiency, and security as well as those of management functionalities. In XML-based network management, we use the security schemes offered by HTTP, HTTP authentication and HTTPS [24] (HTTP over SSL [25]). In management systems, it is sufficient to guarantee the simple security, transparent identification and authentication [4]. Also, the communication security between the XBM manager and the XBM agent is guaranteed by the HTTPS. That is, XML-based network management supports simple security level provided by the HTTP, and this security level is sufficient for management systems. XML can complement and solve the scalability and efficiency problems of SNMP.

As a result, the architecture consisting of the XBM manager and XBM agent takes the maximum advantages of applying XML technologies to the network management. Besides, the XML-based network management complements the weakness of SNMP security through the HTTP authentication and HTTPS protocol. The network overhead is reduced, management processing power is somewhat shifted to the XBM agent, and the XBM manager can manage more agents. So the XBM manager can solve scalability problems with SNMP. However, as this architecture basically assumes the XBM agent, the manager can manage the new XBM agents, but not the existing SNMP agents.



We designed the XBM manager in accordance with XML-based NM for the compensating of weakness of SNMP. This figure illustrates the structure of an XBM Manager. The basic components processing management functionalities are Device Configuration Manager, Polling Engine, Logging Manager, Notification Handler, Event Reporter, and Analyzer. The Device Configuration Manager is a module to set the configuration of a managed device, the Polling Engine is a module to obtain device management information, the Logging Manager logs the necessary data and analyzes the log data stored in DB tables per the administrator's request. The Device Notification Handler receives notifications from the managed devices and stores the notifications to the DB tables and sends a meaningful notification to the Event Reporter. The Event Reporter receives the notifications from the Device Notification Handler and generates appropriate events and sends them to administrators by an email or a pager, etc. The Analyzer is a module to analyze the gathered management information.

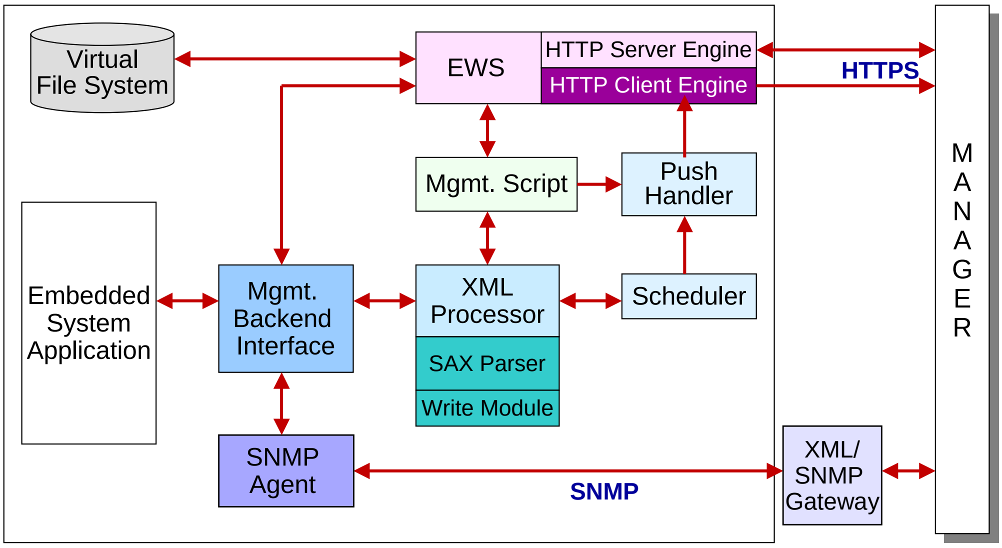
A Web Server is used to provide administrators with a Web-MUI and for receiving requests from the Management Station and passing them to the management functionality modules through the Management Script. Also, the Web Server is used for receiving asynchronous messages for notifications from the XBM Agent over HTTPS. The Web Client plays a role of the interface module of device and exchanges synchronous management information with the XBM Agent. The DB is used to store management information for long-term analysis. The management functionality modules such as the Device Configuration Manager, Polling Engine, etc., use the XML Parser and Translator to implement the management application functions because management information is represented in XML data. These functions include filtering, logging, and collecting data from multiple XBM agents. The management functionality modules use the DOM Interface dealing with the management information formatted the XML Schema, interacting with the DB.

There are three typical information flows within the XBM manager. The first data flow is the management request from the Management Station to the Web Server, and the Web Server calls the Management Script and selects the proper management module. If the management function is to configure a device, the Management Script calls the Device Configuration Manager, and then the Device Configuration Manager sends the request to the XBM agent through the Web Client, and then the result returns to the Management Station in reverse order. If the result needs to be stored for later analysis, the Device Configuration Manager stores them to the DB through the DOM Interface. This flow is used to unify a Web-based element user interface into a Web browser interface without any additional management logic.

Secondly, when the XBM agent sends a notification to the administrator, the information travels in the following order. The XBM agent sends notification through HTTPS. Then, the Web server receives the notification, and calls Notification Handler through Management Script. The Notification Handler sends the specific event to the Event Reporter for generating the appropriate event and stores the other notification for later analysis to DB through the DOM Interface. Also, the XBM agent distributes management information on schedule to the XBM manager. The XBM agent sends it through the Web Server, and the Management Script interacting with the Web Server calls the Polling Engine, then the Polling Engine stores the information to the DB through the DOM and the DB.

The last data flow from the Management Station, through the Web Server, the Management Script, the DOM Interface, to the DB is used to generate a long-term analysis report. For example, the Web Server first calls the Analyzer through the Management Script, then the Analyzer searches for data from the DB using the DOM Interface. After processing the data by filtering, sorting, and correlating, it finally deduces the result. The analysis result is shown to the administrator. The XBM manager can communicate with SNMP agents as well as XBM agent for integrated management. Therefore, the XML/SNMP gateway is necessary.

Design of XBM Agent



APNOMS 2002 - 12 - DP&NM Lab. POSTECH

This slide illustrates the architecture of an XBM agent. The XBM agent includes an Embedded Web Server (EWS) [26] as a basic component. The components added to the EWS are the XML Handler having XML parser and SAX API, the Push Scheduler, and the HTTP Client Engine. In our previous work [3], we showed the use of DOM and XPath for handling the XML document. In the context of network management, DOM and XPath are good for processing XML document for accessing and filtering.

In our previous work, the device embedded XBM agent was Linux server, so there was no problem with the resource. However, supporting DOM and XPath needs a lot of memory resources in the device. If we access a part of an XML document, the DOM tree of the whole XML document is loaded into the memory. This wastes the resources in the device. As embedded systems such as network devices have relatively low computing resources [26], we must reduce memory and processing resources. Therefore, we consider Simple API for XML (SAX) instead of DOM. The code size and executable memory size of SAX is smaller than DOM. As the SAX is an event-driven mechanism for accessing and processing the XML document, there is no need to load the whole XML tree to the memory. Therefore, the SAX is much lighter than the DOM from the aspect of resources. However, the access method of SAX is serial and read-only, and we must provide a writing mechanism, so we add a writing module. Because we set more value on low resource requirements, we selected the SAX.

The HTTP Client Engine sends asynchronous messages to the XBM Manager to notify of alarms and distribute management data according to the schedule. The XML Parser parses the XML document, and SAX API selects the specified node when parsing, and reads management data. In order to send up-to-date information, the XBM agent gathers information from the Management Backend Interface. SAX has no writing functionality, so the Writing Module updates the node value for the selected node through the Management Backend Interface before replying to the XBM Manager. The Scheduler manages subscription information and the schedule for the distribution of the management information and the Push Handler receives the request from the Scheduler and sends the scheduled data to the XBM manager through the HTTP Client at the scheduled time. Also, the Push Handler sends a notification generated in the XBM agent, which is sent to the XBM manager through the HTTP Client.

The SNMP agent is also equipped to the network device, and uses the same Management Backend Interface for reducing memory resource. The XBM manager communicates with the SNMP agent through the XML/SNMP gateway.

Concluding Remarks

- Summarized the weaknesses of SNMP-based network mgmt
- Examined the advantages of applying XML technologies (such as DOM, XPath, XSLT, SAX) to network management
- Proposed approaches to integrate SNMP and XML
  - Several combinations of SNMP and XML management paradigms can exist
  - Presented each approach and proposed how XML can be applied to solve the shortcomings of SNMP-based network management
- Also, proposed the design architectures of XML-based manager and XML-based agent for XML-based management paradigms
- Future work
  - Complete the current implementation of an XBM agent and an XBM manager
  - validate and confirm the effectiveness of XML-based management

APNOMS 2002 - 13 - DP&NM Lab. POSTECH

In this paper, we discussed the weaknesses of SNMP-based network management and examined the advantages of applying XML technologies, such as DOM. XPath, XSLT, SAX, etc., to network management. We also proposed several approaches to compensate for the drawbacks of SNMP: scalability, efficiency, security, and hardness of development, by applying XML technologies to network management. As the SNMP manager and SNMP agent are replaced with the XBM manager and XBM agent by stages, we complemented the weaknesses of SNMP and gained the best advantages of the XML-based network management. We have also proposed detailed design architectures of XBM manager and XBM agent that can be used in XML-based network management.

We are currently developing an XBM manager and an XBM agent based on our proposed design. The XBM manager uses Xerces and Xalan, XML package of Apache group [27] using Java technologies, and the XBM agent uses the Expat XML parser [28] and SAX API for managing IP network access devices on an embedded Linux OS. Our future work will validate and confirm the effectiveness of XML-based management through a complete implementation.

References

[1] J. Case, M. Fedor, M. Schoffstall, and J. Davin (Eds.), “A Simple Network Management Protocol (SNMP)”, RFC 1157, IETF, May 1990.

[2] Tim Bray, Jean Paoli and C. M. Sperberg-McQueen, “Extensible Markup Language (XML) 1.0”, W3 Recommendation REC-xml-19980210, February 1998.

[3] H.T. Ju, M.J. Choi, S.H. Han, Y.J. Oh, J.H. Yoon, H.J. Lee, and J.W.K. Hong, “An Embedded Web Server Architecture for XML-based Network Management”, In Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2002), Florence, Italy, April 2002, pp.1~14.

[4] J.P. Martin-Flatin, “Web-Based Management of IP Networks and Systems”, Ph.D. thesis, Swiss Fed. Inst. of Technology, Lausanne (EPFL), October 2000.

[5] ISO 8879, “Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)”, August 2001.

[6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, L. Masinter, P. Leach and T. Berners-Lee, “Hypertext Transfer Protocol - HTTP/1.1”, RFC 2616, IETF HTTP WG, June 1999.

[7] D. Raggett, A. Le Hors, I. Jacobs, “HTML 4.01 Specification”, IETF HTML WG, http://www.w3.org/TR/html401, December 24 1999.

[8] W3 Consortium, “XML Schema Part 0,1,2”, W3 Consortium Recommendation, May 2001.

[9] W3 Consortium, “Document Object Model (DOM) Level 1 Specification”, W3 Consortium Recommendation, October 1998.

[10] W3 Consortium, “Simple API for XML Version 2.0”, W3 Recommendation, W3 Consortium, November 1999.

[11] W3 Consortium, “Extensible Stylesheet Language (XSL) Version 1.0”, W3 Consortium Candidate Recommendation, November 2000.

[12] W3 Consortium, “XSL Transformations Version 1.0”, W3 Consortium Recommendation, November 1999.

[13] W3 Consortium, “XML Path Language (XPath) Version 1.0”, W3 Recommendation, W3 Consortium, November 1999.

[14] M.T. Rose, “The Simple Book: an Introduction to Networking Management”, Revised 2nd edition. Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[15] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (Eds.), “Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)”, RFC 1902, IETF, January 1996.

[16] ITU-T Recommendation X.690, “Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)”, ITU, Geneva, Switzerland, July 1994.

[17] R. Sprenkels and J.P. Martin-Flatin, “Bulk Transfers of MIB Data”, The Simple Times, 7(1):1–7, March 1999.

[18] A.S. Rao and M.P. Georgeff, “Modeling rational agents within a BDI-architecture”, In R. Fikes and E. Sandewall (Eds.), Proc Knowledge Representation and Reasoning (KR&R-91), San Mateo, CA, USA, April 1991, pp. 473–484. Morgan Kaufmann, 1991.

[19] D. Levi, P. Meyer, and B. Stewart (Eds.), “SNMPv3 Applications”, RFC 2573, IETF, April 1999.

[20] J. Schönwälder and F. Strauss, “Next Generation Structure of Management Information for the Internet”, In R. Stadler and B. Stiller (Eds.), Active Technologies for Network and Service Management. Proc. 10th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM’99), Zurich, Switzerland, October 1999. LNCS 1700:93–106, Springer Berlin, Germany, 1999.

[21] DMTF, “XML As a Representation for Management Information - A White Paper. Version 1.0”, September 1998, http://www.dmtf.org/spec/xmlw.html/.

[22] Bouret, Ronald, “ XML and Databases”, September, 1999, http://www.rpbourret.com/xml/XMLAndDatabases.htm/.

[23] Y. J. Oh, H. T. Ju, M. J. Choi, J. W. Hong, “Interaction Translation Methods for XML/SNMP Gateway”, Submitted to DSOM 2002, Montreal Canada, October 2002.

[24] Apache-SSL, http://www.apache-ssl.org/.

[25] W. Stallings, “SSL: Foundation for Web Security”, The Internet Protocol Journal, 1(1):20–29, 1998.

[26] M.J. Choi, H.T. Ju, H.J. Cha, S.H. Kim, and J.W.K. Hong, “An Efficient and Lightweight Embedded Web Server for Web-based Network Element Management”, In Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2000), Hawaii, USA, April 2000, pp. 187~200.

[27] Apache XML Project, http://xml.apache.org/.

[28] J. Clark, Expat XML Parser, http://expat.sourceforge.net/.