Integrated Enterprise Management Using WBEM/SNMP Gateway

So-Jung Lee*, Hong-Taek Ju** and James W. Hong*

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

Abstract
The TMN framework has been developed for the operations and management of telecommunication networks and the SNMP framework for that of the Internet networks. Neither TMN or SNMP was originally developed for the management of computing resources (i.e., computing servers and applications). Recently, WBEM has been proposed to manage enterprise computing and networking resources. Most computing resources that are currently being developed and deployed are WBEM compliant. However, most of networking resources are not yet WBEM compliant. In order to manage network devices that are equipped with SNMP only using the WBEM framework, a WBEM/SNMP gateway is necessary. In this paper, we provide an overview of the standardization effort of WBEM. We then propose a WBEM-based integrated server and network server management system that connects with existing SNMP supported devices through a WBEM/SNMP gateway. We analyze the requirements for a WBEM/SNMP gateway, and present a design for an overall WBEM-based management system and a WBEM/SNMP gateway that satisfies the requirements. The WBEM/SNMP gateway provides two mechanisms; a specification translation and an interaction translation. Finally, we present our WBEM/SNMP gateway prototype and WBEM Manager that have been implemented for validation.

Keywords: WBEM/SNMP Gateway, WBEM-based Management, Enterprise Management, WBEM, SNMP

1. Introduction
The TMN framework has been used to operate and manage telecommunication networks for the last two decades. On the other hand, the SNMP framework has been mainly used to operate and manage Internet networks since late 90’s. At the same time, there has been a tremendous growth in the deployment and use of computers and applications in enterprises, businesses and entertainment. The size and complexity of managing these computing and networking resources have also increased considerably. Unfortunately, neither TMN or SNMP was originally developed to manage computing resources and is found to be inadequate to manage them. Further, the management of these has been done in an ad hoc, proprietary fashion for many years. The enterprise management industry has recognized this problem and an industry consortium called Distributed Management Task Force (DMTF) has been standardizing Web-
based Enterprise Management (WBEM) to provide a set of standards for managing enterprise computing and networking resources [1].

WBEM is a set of management and Internet standard technologies developed to unify the management of distributed computing environments, facilitating the exchange of data across disparate technologies and platforms. WBEM defines management information in Common Information Model (CIM) [2], encodes the message in CIM-XML and transports it using the HTTP protocol. WBEM also includes additional technologies such as CIM Query Language [3], WBEM Discovery using SLP [4] and WBEM URI mapping [5]. Like Microsoft server systems, Sun server systems and Cisco network devices load WBEM agents, WBEM is expected to diffuse. However, till now, most legacy devices were entrained with an SNMP agent. Hence, integrating protocols between different domains is necessary.

In order to manage enterprise computing and networking devices that are not WBEM compliant (i.e., WBEM servers are not installed) but only SNMP agents are installed, a WBEM/SNMP Gateway is required to manage SNMP supported devices using the WBEM framework. WBEM has specified a functional module called SNMP Provider to support the integration of SNMP devices using the WBEM framework. However, the currently available implementations offer such functionality within a single system. That is, the WBEM server and SNMP agent are within the same device and do not provide the capability of supporting multiple remote devices.

For this reason, we have developed a WBEM/SNMP Gateway that is flexible and scalable. The WBEM/SNMP Gateway provides specification translation and interaction translation mechanisms to support interoperability between WBEM and SNMP domains. For specification translation, we have developed a mapping method to convert from SNMP MIB to CIM MOF. For interaction translation, we have developed a method that converts CIM operations into SNMP operations. We also present our prototype implementations of WBEM/SNMP gateway and WBEM Manager for validation.

The remainder of this paper is organized as follows. A general overview of WBEM is provided in Section 2 along with an introduction to commercial efforts and open source projects of WBEM implementations. In Section 3, we present the design requirements for a WBEM/SNMP Gateway. Also, we present the specification translation and interaction translation algorithms of a WBEM/SNMP Gateway. In Section 4, we briefly explain the implementation of our WBEM/SNMP Gateway prototype and WBEM Manager. Finally, we summarize our work and discuss possible future work in Section 5.

2. WBEM-based Enterprise Management

In this section, we explain the technologies that are basis of WBEM. Then, we introduce industry implementations and open source projects of WBEM.


2.1 WBEM Standards

WBEM is an initiative of DMTF and it includes a set of technologies that enables the interoperable management of an enterprise network. The DMTF has developed a core set of standards that make up WBEM, which includes a data model, the CIM standard; an encoding specification, CIM-XML encoding specification; and a transport mechanism, CIM operations over HTTP.

The CIM specification is the language and methodology for describing management data. CIM is an object-oriented schema for modeling the managed objects. The CIM schema can be divided into three areas; the core model, the common model and the extension model. First, the core model captures notions that are applicable to all areas of management. Second, the common model is an information model that captures notions that are common to a particular technology. For example, it includes the model for systems, applications, networks and devices. Last, the extension model represents technology-specific extensions of common models. For example, the NT_LightPath_Service class can inherit from CIM_NetworkService Class. CIM can be written in Management Object Format (MOF), XML or UML.

The CIM-XML encoding specification defines XML elements, written in Document Type Definition (DTD) that can be used to represent CIM classes and instances. The CIM operations over HTTP specification [6] defines a mapping of CIM Operations in XML over HTTP that allows implementations of CIM to interoperate in an open, standardized manner. Therefore, in the WBEM architecture, the management information is described by the CIM schema, converted to XML, and finally embedded in an HTTP payload to transport to the target node.

![Figure 1. WBEM System Architecture](image)

Figure 1 illustrates the WBEM architecture, which includes a WBEM Client, and WBEM Server. WBEM Server has CIM Object Manager (CIMOM) which is a central component that routes information about objects and events between components. CIMOM responds to the operations defined in the “CIM operations” specification such as create, modify, and delete. It also checks the syntax and semantic of the messages, and provides security. Providers are so-called instrumentation agents. Namely, Providers actually obtain the information from the resources and forward it to the CIMOM. A WBEM Client is
commonly represented as the management application, and it can get the information by sending a request message to the CIMOM instead of directly accessing the providers.

The DMTF working group is also standardizing additional technologies for WBEM such as WBEM CIM Query Language [3], WBEM Discovery using SLP [4] and WBEM URI mapping [5].

- **WBEM Discovery using SLP**

  The Service Location Protocol (SLP) provides a flexible and scalable framework for providing clients, represented by User Agents, with access to information about existence, location and configuration of services, represented by Service Agents. WBEM Servers acting as Service Agents advertise their services, and WBEM Clients acting as User Agents query for the WBEM Servers. The specification for WBEM Discovery using SLP is defined for WBEM Servers to advertise their service access point and capabilities using SLP.

- **WBEM URI Mapping**

  WBEM Uniform Resource Identifiers (URI) Mapping is used by WBEM protocols for mapping between the CIM Naming and the URI format. The WBEM URI is a compact string of characters to identify a CIM element. The WBEM URI is defined as follows.

  ```
  Scheme://[@]<userinfo>[@]<host>["":<port>]"/<namespace>"/:<model path>
  ```

- **CIM Query Language**

  The CIM Query Language has been widely used in CIM operations over HTTP specification, using the CIM Events Model and the CIM Policy Model. Query semantics must include instance property projection (e.g., a SQL SELECT clause), a range (e.g., A SQL FROM clause) and may include predicate logic (e.g., a SQL Where clause).

### 2.2 Industry Implementations

Generally, WBEM is provided in the form that each vendor’s network management system solution and application supports. In the case of IBM, WBEM is provided by its representative network and systems management solution suite called Tivoli, including “Tivoli NetView”, “Tivoli Enterprise”, “Tivoli Manager for IBM NWays” and “Tivoli Cross-Sight”. Cisco utilizes CIM in the enterprise network management tool called CiscoWorks2000. HP provides the WBEM solutions that include WBEM Services, WBEM Providers, HP WBEM Client and HP WBEM SDK. Microsoft provides a Web-based Enterprise Management System that is embedded in the MS-windows OS called WMI. Sun provides Solaris WBEM Services and Sun WBEM SDK. Both Microsoft WMI and Sun Solaris include a SNMP Provider. We specifically describe both the Microsoft and Sun SNMP Providers below.

#### 2.2.1 Microsoft WMI SNMP Provider

WMI [7] is a WBEM Server that is embedded in the Microsoft Windows operating system. It is a WBEM Server that follows the WBEM standard. WMI has CIMOM and Object Repository, and a
CIMOM can communicate with database applications or WEB Browser. A Win32 Provider acts as an intermediate between OS components and applications in the Win32 Environment. Likewise, the Registry Provider can access the Registry data. WMI includes an SNMP Provider that can read or write SNMP MIB Object variables. The SNMP Provider can map each object data to the properties of CIM Class instances. Also, an SNMP trap can be automatically mapped to a WMI event.

WMI also includes the smi2smir utility that translates a MIB file to a MOF file. The translation result can be stored in the SNMP Module Information Repository (SMIR) that is SNMP’s schema database. A MIB can be converted to the CIM schema using several Macros. That is, macros such as a NOTIFICATION-TYPE Macro, an OBJECT-TYPE Macro, a TEXTUAL-CONVENTION Macro and a Trap-TYPE Macro determine the MIB to MOF mapping method. Especially, the OBJECT-TYPE macro includes an essential item to represent the basic characters of a MIB object.

One of the weaknesses of the WMI SNMP Provider is that the translation result depends on the WMI. A function like ‘ToInstance’ is not declared in the other WMI implementations. Therefore, the MOF file that is generated by smi2smir must be modified in order to be compiled by another system’s MOF compiler. Also, the WMI SNMP Provider does not provide open source code for the interaction translation. Moreover, the WMI SNMP Provider is limited to the Windows OS, and it is not provided for UNIX or LINUX.

2.2.2 Sun Solaris SNMP Provider

Sun Solaris includes a WBEM SNMP Provider that enables WBEM Services to deliver information. Sun Solaris also includes a mib2mof utility that converts a MIB file to a MOF file. The mib2mof utility reads the MIB file as an input file and generates one or more MOF files. A MOF file includes CIM classes that represent the SNMP groups or rows. The mib2mof basically uses the standard Qualifiers that are provided by DMTF. Sun Solaris’s SNMP Provider has a drawback, because no open source support is provided for interaction translations and also it is limited to hardware using Solaris, and it is not provided for Microsoft Windows.

2.3 Open Source Projects

This subsection presents a brief overview of several open source projects related to WBEM.

2.3.1 OpenPegasus

OpenPegasus [8] is one of the open source WBEM implementations, which is extended by us for developing our WBEM/SNMP gateway system. The Open Group (TOG) actively leads in the development of this open source with plenty of technical documents. Pegasus also has Client and Server, which communicates with CIM/HTTP. Also, the CIM Server can access CIM schemas which are in the repository, and can invoke the provider to manage the management information. Pegasus has the cimmo
utility that acts as an MOF Compiler. Pegasus is written in C++ code, and the repository is written in XML.

2.3.2 WBEM Services

WBEM Services [8] and Sun WBEM SDK are also open source WBEM implementations which are being developed under Sun’s auspices. WBEM Services has a JAVA-based CIMOM and offers a JAVA API for a client and provider. When a client tries to get or modify information of a management object through the client API, the CIMOM responds to the client’s request by interacting with the CIM repository or the related provider. Various management objects can be SNMP Objects, Native Objects or any other Objects.

The CIMOM also performs syntactic and semantic verification. The syntactic verification discovers errors such as the omission of a semicolon or a brace, and the semantic verification finds any logical errors in the program. The WBEM Services’ providers exist as MOF files. These providers represent the systems, processes and resources such as CPU cycles, memory and so on. WBEM Services also provide a security service such as authentication and authorization. For users to be authenticated, the client users need to provide the user ID and password for the WBEM server. Also the WBEM server has an Access Control List (ACL) which has a list of users that are allowed to access to the WBEM server.

2.3.3 OpenWBEM, SBLIM and SNIA CIMOM

OpenWBEM is another open source project, which is maintained by Vintela, Novell and other volunteers. It is written in C++, and suitable for commercial and non-commercial applications. Vintela VMX, Novell/SUSE SLES 9 and Apple Remote Desktop 2 are the commercial products using OpenWBEM.

Standard Based Linux Instrumentation Manageability (SBLIM) [10] is an IBM open source project, intended to enhance the manageability of GNU/Linux systems. It can be used in Pegasus, OpenWBEM and SNIA CIMOM. Many of the SBLIM Linux providers work with the OpenWBEM NPI & CMPI provider interfaces.

SNIA CIMOM is driven by SNIA (Storage Networking Industry Association). SNIA CIMOM is a JAVA daemon and has no repository. SNIA CIMOM has had almost no research activity after 2002.

3. Design of WBEM/SNMP Gateway

In this section, we first analyze the requirements of the WBEM/SNMP Gateway. Based on the requirements, we present the design of the architecture of an overall management system using the WBEM/SNMP Gateway. We also present the specification translation and interaction translation mechanisms for the WBEM/SNMP Gateway.
3.1 Requirements Analysis

The following are the functional and nonfunctional requirements for WBEM/SNMP Gateway.

- The Manager’s ability includes topology management, monitoring and analysis.
- WBEM/SNMP Gateway has to provide a mechanism to integrate existing SNMP devices.
- WBEM/SNMP Gateway has to provide the mechanism to initiate the connection to export the Trap information, in case an error is generated from SNMP devices.
- WBEM/SNMP Gateway has to provide a security mechanism.
- WBEM/SNMP Gateway has to provide high scalability and flexibility.
- Web-based User Interface has to be provided for administrators to access anytime and anywhere.

3.2 Architecture Design

This subsection presents the design architecture of a WBEM-based management system and the WBEM/SNMP Gateway.

3.2.1 WBEM-based Management System Architecture

The architecture is based on the manager-agent architecture, and the gateway provides interoperability between a manager and an agent. In the viewpoint of WBEM-based architecture, WBEM Manager is a WBEM Client, and a WBEM/SNMP Gateway is a WBEM Server. Administrators can access the WBEM Manager through a management application. The WBEM Manager and WBEM/SNMP Gateway communicate with the XML/HTTP, and the WBEM/SNMP Gateway and SNMP Agents communicate with SNMP.

3.2.2 WBEM/SNMP Gateway Architecture

Here, we describe the architecture of our WBEM/SNMP Gateway. As shown in Figure 2, the WBEM/SNMP Gateway serves two functions: specification translation and interaction translation. Specification translation converts the SNMP MIB file to a CIM MOF file, and interaction translation converts a CIM Operation to an SNMP Operation. For specification translation, the WBEM/SNMP Gateway includes a MIB2MOF Translator that converts the management information defined in MIB into MOF. Then, the MOF compiler accepts MOF input and populates the CIM Repository.

For interaction translation, a WBEM/SNMP Gateway includes the following components. The HTTP Server in the WBEM/SNMP Gateway gets the message from the WBEM manager, and the HTTP Client is required to send trap information to the WBEM manager. The CIM-XML Decoder/Encoder converts the message, and delivers it to the CIMOM. First, the CIMOM attempts to retrieve the data requested by an administrator from the repository. If the data is not in the repository, the CIMOM obtains it from the provider. The data is then mapped to the CIM operation and the results are returned to the WBEM manager.
The Provider Manager manages various providers including an Instance Provider, a Method Provider, an Association Provider, and an Indication Provider. The Instance Provider creates/modifies/deletes an instance or gets/sets a property value. The Method Provider plays a role in invoking methods such as the kill operation. The Association Provider can access an association or a reference class. The Indication Provider translates the occurrence of an event into a CIM indication and sends the indication to the CIMOM for further processing and delivery.

An SNMP Provider is served by a registering process like other providers. The SNMP Provider can get the CIM operation from CIMOM and forward it to the appropriate CIM Request Handler. The CIM Request Handler calls the SNMP Operation that maps to each CIM Operation, and finally the SNMP stack can send the SNMP request to the SNMP agent. If the SNMP agent sends the MIB variable back to the WBEM/SNMP Gateway, then the WBEM manager can receive the result. The Indication Service processes the creation, modification and deletion operations for the indication subscriber data, which includes the list of clients who subscribe to get the indication.

We have considered and incorporated security in three areas. Managers and agents can communicate more securely by using HTTP over SSL (HTTPS), and clients can pass a user ID and a password for user authentication. Also a user can be granted the appropriate permission (i.e., read and write) to access a target namespace.
3.3 Specification Translation

For specification translation, a MIB2MOF Translator can convert an SNMP SMI Datatype to a CIM Datatype. As shown in Table 1, the Object Datatype which is the SNMP MIB’s SYNTAX information can map to a CIM Datatype.

<table>
<thead>
<tr>
<th>SNMP SMI Datatype</th>
<th>SNMP Ver.</th>
<th>CIM Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>v1</td>
<td>Sint32</td>
</tr>
<tr>
<td>OCTET STRING</td>
<td>v1</td>
<td>String</td>
</tr>
<tr>
<td>OBJECT IDENTIFIER</td>
<td>v1</td>
<td>String</td>
</tr>
<tr>
<td>IpAddress</td>
<td>v1</td>
<td>String</td>
</tr>
<tr>
<td>Counter</td>
<td>v1</td>
<td>Unit32</td>
</tr>
<tr>
<td>Gauge</td>
<td>v1</td>
<td>Unit32</td>
</tr>
<tr>
<td>TimeTicks</td>
<td>v1</td>
<td>Unit32</td>
</tr>
<tr>
<td>Opaque</td>
<td>v1</td>
<td>Sint8[ ]</td>
</tr>
<tr>
<td>DisplayString</td>
<td>v1</td>
<td>String</td>
</tr>
<tr>
<td>NetworkAddress</td>
<td>v1</td>
<td>String</td>
</tr>
<tr>
<td>Counter64</td>
<td>v2</td>
<td>Unit64</td>
</tr>
<tr>
<td>Integer32</td>
<td>v2</td>
<td>Sint32</td>
</tr>
<tr>
<td>Gauge32</td>
<td>v2</td>
<td>Unit32</td>
</tr>
<tr>
<td>Unsigned32</td>
<td>v2</td>
<td>Unit32</td>
</tr>
<tr>
<td>TruthValue</td>
<td>v2</td>
<td>Sint32</td>
</tr>
<tr>
<td>BITS</td>
<td>v2</td>
<td>String</td>
</tr>
</tbody>
</table>

Table 1. The mapping table between SNMP SMI Datatype and CIM Datatype

As described in Table 2, SNMP MIB’s ACCESS, STATUS and DESCRIPTION information is defined in CIM Qualifier. The CIM Qualifier is included in the CIM specification which is standardized by the DMTF.

<table>
<thead>
<tr>
<th>OBJECT TYPE MACRO</th>
<th>CIM Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS</td>
<td></td>
</tr>
<tr>
<td>Read – Only</td>
<td>Read</td>
</tr>
<tr>
<td>Read – Write</td>
<td>Read, Write</td>
</tr>
<tr>
<td>Write-only</td>
<td>Write</td>
</tr>
<tr>
<td>Not-accessible</td>
<td>(write nothing)</td>
</tr>
<tr>
<td>STATUS</td>
<td></td>
</tr>
<tr>
<td>Mandatory</td>
<td>Required</td>
</tr>
<tr>
<td>Optional</td>
<td>(write nothing)</td>
</tr>
<tr>
<td>Obsolete</td>
<td>(write nothing)</td>
</tr>
<tr>
<td>Deprecated</td>
<td>(write nothing)</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Description</td>
</tr>
</tbody>
</table>

Table 2. The relationship between OBJECT TYPE MACRO and CIM Qualifier

In this paper, we define inheritance relationship between CIM classes for RFC1213-MIB by making new CIM extension schemas with the CIM model’s CIM core schema. As described in Figure 3, we define a new extension class named mib-2 class and it inherits from the ManagedSystemElement Class which is the child class of a ManagedElement Class. Also, RFC1213-MIB’s each group which is defined as a class can be inherited by mib-2 class.
RFC1213-MIB’s classes includes RFC_1213_MIB_system class, RFC_1213_MIB_interfaces class etc. That is, the group node of each SNMP MIB tree can be converted to the CIM Class name, and each leaf node can be converted to each class’s property. Also, we can exchange the SNMP MIB’s Datatype and OBJECT TYPE MACRO to the CIM Datatype and Qualifier. Because the resulting MOF is only defined according to the DMTF standard, it could be used in any other implementations without dependency on a vendor or hardware.

3.4 Interaction Translation

Next, we show the interaction translation mechanism of the WBEM/SNMP Gateway. Table 3 describes the method used to map each CIM operation to the SNMP operation.

<table>
<thead>
<tr>
<th>Function</th>
<th>CIM Operation</th>
<th>SNMP Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Read</td>
<td>GetClass, EnumerateClasses, EnumerateClassNames, GetInstance, EnumerateInstances, EnumerateInstanceNames, GetProperty</td>
<td>SNMP Get, SNMP GetNext</td>
</tr>
<tr>
<td>Basic Modify</td>
<td>SetProperty</td>
<td>SNMP Set</td>
</tr>
<tr>
<td>Schema Manipulation</td>
<td>CreateClass, ModifyClass, DeleteClass</td>
<td>-</td>
</tr>
<tr>
<td>Instance Manipulation</td>
<td>CreateInstance, ModifyInstance, DeleteInstance</td>
<td>SNMP Set</td>
</tr>
<tr>
<td>Association Traversal</td>
<td>Associators, AssociatorNames, References, ReferenceNames</td>
<td>-</td>
</tr>
<tr>
<td>Qualifier Declaration</td>
<td>GetQualifier, SetQualifier, DeleteQualifier, EnumerateQualifier</td>
<td>-</td>
</tr>
<tr>
<td>Query</td>
<td>ExecQuery</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3. The mapping table for interaction translation
Among the basic reading command, EnumerateInstances, GetInstance and GetProperty can get the property information which are leaf nodes of the MIB tree. The difference between EnumerateInstance, GetInstance and GetProperty is merely this. EnumerateInstances can get the property information from all the instances that are created. Assuming that each agent creates its own instance, we can retrieve all the management information from the agent through the EnumerateInstance operation. However, GetInstance can get only one instance that includes a specific key value. As the key value is an IP address, GetInstance operation can get the management information from only one SNMP Agent. On the other hand, the GetProperty operation can get one property value from one instance. Therefore EnumerateInstances operations can be converted to SNMP GetNext operation, and GetInstance or GetProperty operation can be replaced by the SNMP Get operation.

The instance manipulation and basic modification commands such as ModifyInstance and SetProperty can modify the property value which is the leaf node in the MIB tree. Therefore the ModifyInstance and SetProperty operations can be converted to the SNMP Set Operation. However, Schema Manipulation functions such as CreateClass, ModifyClass or DeleteClass operations and the Basic Read operation functions such as GetClass, EnumerateClasses, EnumerateClassNames or EnumerateInstanceNames operations do not access the SNMP Agent. Instead they directly query the repository and get the value. Therefore there is no SNMP Operation to be converted. Also, Association traversal, Qualifier declaration and Query function have no suitable SNMP Operation to be replaced.

4. Prototype Implementation

From the early stages of our WBEM/SNMP Gateway development, we planned to extend an existing WBEM implementation. To choose the most excellent WBEM implementation, we performed Benchmark testing on existing WBEM implementations. Based on the results of this testing, we implemented our prototype of a WBEM/SNMP gateway by extending Pegasus. We also implemented a prototype of the WBEM Manager.

4.1 Implementation of WBEM/SNMP Gateway

For the specification translation function of a WBEM/SNMP Gateway, we implemented the MIB2MOF Translator. By a MIB2MOF Translator, all management information that is defined as MIB can be converted to the MOF format. We then compiled the MOF file through the MOF compiler and registered the management information to the repository. Through the MIB2MOF Translator and the MOF compiler, the WBEM/SNMP Gateway has flexibility to support any SNMP MIB. Also the MOF files produced by the MIB2MOF Translator can be used in any other WBEM implementations.

To provide the interaction translation function, we implemented the backend interface of an SNMP provider, that is, the SNMP Provider module. The SNMP Provider included in the WBEM-based
management system gets the management information of RFC1213-MIB through the NET-SNMP API. The SNMP Provider is an Instance Provider and we implemented each function that is required for the Instance Provider.

### 4.2 Implementation of WBEM Manager

For the prototype implementation of WBEM Manager, we implemented a Web User Interface for an administrator to access and manage SNMP Agents through the WBEM/SNMP Gateway anytime anywhere. A WBEM Manager can perform the CIM Operations which are listed in the menu as a WBEM Client.

CIM Operations such as CreateInstance, GetInstance, GetProperty, EnumerateInstances and SetProperty can be performed after we input parameters such as Agent’s IP Address, property name or new property value. For example, if we want to perform a GetProperty operation, we need to select not only the Agent’s IP Address, but also the property name. If we want to perform a SetProperty operation, we need to additionally input the new property value for modification.

### 5. Concluding Remarks

In this paper, we briefly reviewed the technologies of the WBEM standard of DMTF, and surveyed industry implementations such as Microsoft WMI and Sun Solaris WBEM. Also, we described the work being done on open source projects such as Pegasus, WBEM Services, OpenWBEM and SNIA CIMOM. We proposed a WBEM-based management system and a WBEM/SNMP Gateway architecture. We also designed a translation method to convert MIB defined information to CIM defined information (specification translation), and an exchanging method to map the CIM operation to the SNMP operation (interaction translation). For validation, we extended the open source software Pegasus and implemented a prototype of a WBEM/SNMP Gateway and a WBEM Manager.

In this paper, we focused on the flexibility of the WBEM/SNMP Gateway. Through the specification translation, any SNMP MIB can be supported by the WBEM/SNMP Gateway. Also, through the interaction translation, the WBEM/SNMP Gateway can manage server and network devices with embedded SNMP Agents.

To validate the scalability of the WBEM/SNMP Gateway that we proposed, we plan to manage more than a hundred SNMP Agents with the WBEM/SNMP Gateway. We also plan to design a lightweight WBEM Agent in the term of a WBEM-based network management.

### References

[9] Sun Microsystems, Java Web Based Enterprise Management (WBEM Services),
[10] IBM, Standard Based Linux Instrumentation Manageability (SBLIM) project,
http://www124.ibm.com/sblim./