Heterogeneous Tools for Heterogeneous Network Management with WBEM

Kenneth Carey & Fergus O' Reilly
Adaptive Wireless Systems Group
Department of Electronic Engineering
Cork Institute of Technology, Cork, IRELAND
Tel. +353 21 4326100   Fax: +353 21 4326625
Email: kcarey@cit.ie, foreilly@cit.ie

Abstract

The rapid roll out of e-commerce applications and the drive for m-commerce applications has caused the boundaries between traditional telecommunications and computer communications and storage to break down. This has lead to heterogeneous networks, mixing traditional voice and new Internet communication and storage technologies. The Web Based Enterprise Management (WBEM) initiative proposes to tackle the problem of managing heterogeneous networks. This paper will discuss the experiences of developing a Heterogeneous Management System with a broad range of heterogeneous tools and toolkits, using WBEM as the unifying principle. We will examine each component of a management system and show how different vendors tools integrate under this.

1. Introduction

Network management today is becoming increasingly difficult because of the need to manage multiple devices each using different network technologies. As a result network administrators are required to deploy numerous management applications and techniques, with no consistent view of the overall system. Since e-commerce and m-commerce networks can and must exponentially grow in size and complexity, network administrators also experience an almost constant learning curve simply to stay up with new developments, rather than improving the network systems.

The WBEM initiative, developed by the Distributed Management Task Force (DMTF), was introduced by a number of companies in 1996 to promote the use of Internet technologies for the management of desktops, servers, and all other Internet-related devices [1]. WBEM standardises the description and use of managed resources/devices in enterprise networks. SNMP dominates the interoperable network-management software market, with companies like EMC, Cisco and 3Com producing SNMP enabled storage devices, switches and routers. CMIP is used mainly for the management of telecommunications network devices, such as switches, repeaters and gateways from companies like Ericsson, Nortel and Motorola.

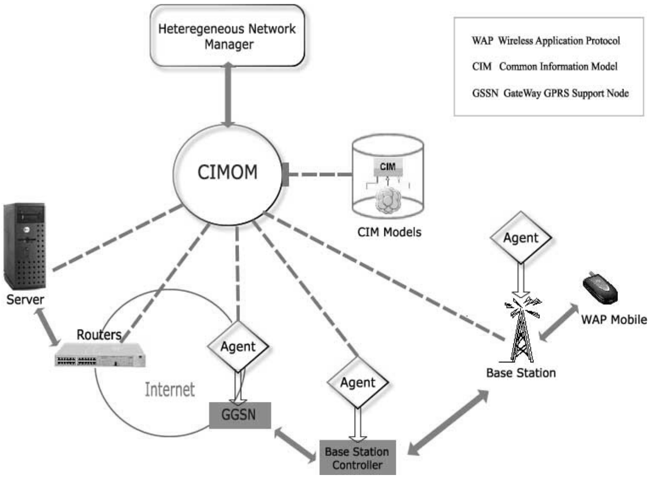

Heteregeneous Network Manager
WAP Wireless Application Protocol
CIM Common Information Model
GSSN GateWay GPRS Support Node
CIMOM
CIM
CIM Models
Server
Routers
Internet
Agent
GGSN
Agent
Base Station Controller
Agent
Base Station
WAP Mobile

Figure 1. Heterogeneous Manager for GPRS/UMTS network

As illustrated in Figure 1, for a GPRS/UMTS Network, the aim of this research project is to develop management architecture and tools, using the WBEM initiative and various heterogeneous technologies, for managing a simulated Heterogeneous Network consisting of SNMP and CMIP agents. This paper will concentrate on the tools and technologies used and any difficulties encountered during their integration into the WBEM system.

2. Web Based Enterprise Management

There are three fundamental requirements for any management environment:

- A data description
- On-the-wire encoding
- Operations to manipulate the data.

The WBEM consists of a data definition format known as the Common Information Model (CIM), an encoding for transport (xmlCIM) and a set of Hyper Text Transfer Protocol (HTTP) operations that define how a management application acquires the management data and how an agent of a managed device responds to the operations.

2.1 Common Information Model

The CIM is a mechanism for modeling managed resources and representing those models in the Managed Object Format (MOF) language [1]. Similar to the Management Information Base (MIB) used to describe SNMP managed resources, it is a textual format for describing management information. Using the CIM and MOF, the components that make up a managed resource or a network of resources can be modeled and viewed in a way similar to that used in an object-oriented software design process.

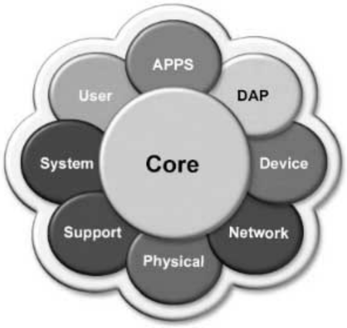


Figure 2. CIM Schemas [2]

The DMTF has proposed a layered class hierarchy that correctly represents manageable elements from a variety of areas. Each management area, such as storage or applications, is represented in a CIM Schema. Figure 2 shows how the existing CIM schemas are conceptually layered. The Core schema, located at the center and other schemas build on each other to represent more specific management areas such as Network and Applications. For example, all managed elements inherit from a class known as a CIM_ManagedElement (see Figure 3).

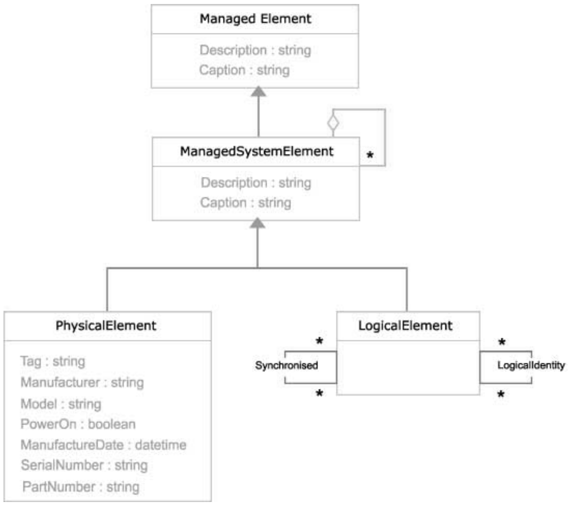


Figure 3. Sample CIM class hierarchy

Once a CIM model and MOF definition are complete, the package is imported into a CIM Object Manager (CIMOM). The CIMOM provides a central repository where clients in a network can go gather information about managed resources within the system. MOF files are imported into the CIMOM, and instances of the classes that relate to a specific managed resource can be created within the CIMOM.

2.2 Transport Encoding

For applications to interoperate with one another it must be possible to represent actual management data in a standard way. Extensible Markup Language (XML) is a markup language for representing information in a standard format. An XML schema is a grammar that describes the structure of an XML document. In WBEM, the XML schema is used to describe the CIM, and both CIM classes and instances are valid XML documents for that schema [3].

2.3 CIM Operations over HTTP

To allow WBEM to operate in an open, standardized manner, the DMTF has defined a mapping of CIM operations onto the Hyper Text Transfer Protocol (HTTP). HTTP, running on top of TCP/IP, can be used for many tasks through the extension of its request methods, error codes and headers. By defining a number of HTTP extension headers for CIM, Client and Server applications are aware that CIM information is being sent. The encoding of CIM to XML favors completeness over conciseness and consequently performance is compromised. Thus the WBEM's reliance on generality as opposed to efficiency is a possible shortcoming.

2.4 Architecture for WBEM Applications

Individual companies provide their own implementation of WBEM, each of which must conform to the specifications, accept the standardized schemas and all associated CIMOMs must be accessible via the standard HTTP operations. Java based implementations are provided by Sun and the Storage Networking Industry Association, in the form of the Sun WBEM SDK [4] and the SNIA CIMOM [5] respectfully. The SNIA CIMOM has been chosen for the development of our WBEM system because of its platform independence and excellent client, cim and provider APIs.

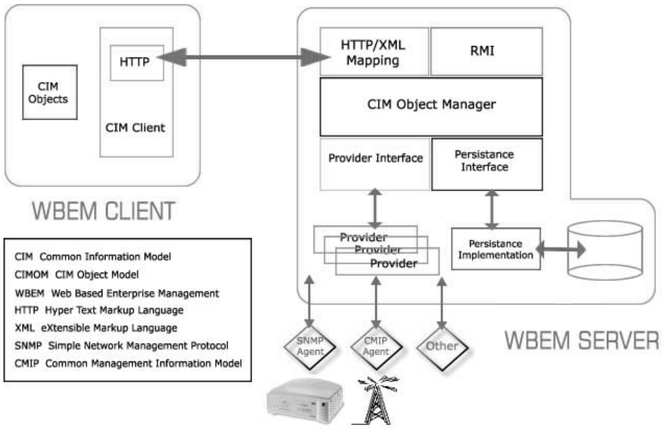

CIM Objects
HTTP
CIM Client
WBEM CLIENT
HTTP/XML Mapping
RMI
CIM Object Manager
Provider Interface
Persistance Interface
Provider
Provider
Provider
Persistance Implementation
SNMP Agent
CMIP Agent
Other
WBEM SERVER
CIM Common Information Model
CIMOM CIM Object Model
WBEM Web Based Enterprise Management
HTTP Hyper Text Markup Language
XML eXtensible Markup Language
SNMP Simple Network Management Protocol
CMIP Common Management Information Model

Figure 4. WBEM Client-Server Architecture

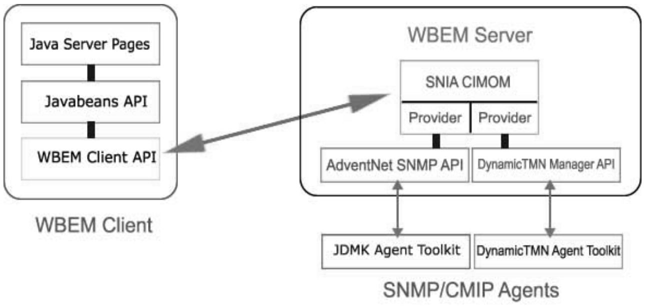


Figure 5. WBEM Client-Server Tools

To build a WBEM enabled system a client–server architecture was designed as illustrated in Figure 4. The core component of the WBEM server is the SNIA CIMOM. The SNIA implementation includes an interface for communication with a WBEM client (HTTP/xml, RMI) and a number of WBEM providers. Server side provider classes are Java classes that use the Provider API to access the CIMOM. Providers allow a client of the CIMOM to gather data from managed resources, such as an SNMP enabled router or a CMIP telecom switch. For example, if the management application requests dynamic data from an SNMP agent the CIMOM uses the provider interfaces to pass the request to the provider class. The provider class performs the necessary SNMP Get operation to retrieve the information from the agent and passes the information to the CIMOM in the form of CIM Java classes. The provider classes therefore act as managers and incorporate SNMP and CMIP functionality. The incorporation of protocol management functionality into the WBEM Server required the integration of SNMP and CMIP management APIs with the WBEM system. The excellent provider concept of the WBEM allowed for these to be seamlessly integrated into the system. The Client application uses the SNIA WBEM client API package to connect and transfer data to the CIMOM and Java Server Pages and Javabean technologies for the development of a web based management console through which the user can access and manage the simulated Heterogeneous Network. Figure 5 outlines the tools incorporated into the design of the WBEM architecture.

3. Experiences of Developing SNMP/CMIP Manager /Agent classes

The Simple Network Management Protocol (SNMP) was created in 1988 as a short-term solution to manage elements in the growing Internet and other attached networks [6]. SNMP is based on the manager/agent model consisting of a manager, an agent, a management information base (MIB) and the network protocol. The manager provides the interface between the human network manager and the management system. The agent provides the interface between the manager and the physical device(s) being managed.

The Common Management Information Protocol (CMIP) is a network management protocol built on the Open Systems Interconnection (OSI) communication model. The related Common Management Information Services (CMIS) defines services for accessing information about network objects or devices, controlling them, and receiving status reports from them [6]. CMIP is responsible for performing the actual manager-agent communication to perform the higher-level operations requested by the CMIS. CMIP MIBs are described using Guidelines for Definition of Managed Objects (GDMO). GDMO is a standard for defining objects in a telecom network in a consistent way.

The development of SNMP and CMIP agents are necessary to simulate heterogeneous network devices. They are therefore separate entities from the WBEM Server and are created using agent toolkits and management APIs. The Java Dynamic Management Toolkit, AdventNet Agent Toolkit and the Monfox DynamicTMN toolkit are rapid prototyping and development tools used for building cross platform Java based SNMP and CMIP agents. Application Programming Interfaces (APIs) provided with these toolkits allow for the development of Java based SNMP and CMIP management classes for performing operations on these agents. These management APIs are integrated into the WBEM server providers to allow the CIMOM to operate an agents dynamic data.

3.1 Java Based SNMP and CMIP Agents

Java Dynamic Management Kit

The Java Dynamic Management Kit [7] is a Java API based on the Java Management Extensions (JMX) specification and is used for designing cross-platform multi-protocol agents. The JDMK is not specific to a certain protocol like SNMP; instead it is architecture for abstracting and remotely accessing data objects within a Java platform.

Development of Java based agents using the JDMK toolkit involved the creation of a MIB, generation of source code for the defined MIB using a mib compiler and finally the instrumentation of the generated code to add desired functionality to the agent. The agents describe system information, such as system name, contact person etc. It was therefore necessary to write a MOF definition and load it into the CIMOM. The MOF and MIB definitions are similar in that both describe a device, using name, location, contact person, status, serial number etc. Instead of returning default values the agents use an Access database to retrieve and update mib variables via JDBC. If a GET operation is performed then a query on the database is made using SQL, likewise if the manager issues a SET operation then the database is updated accordingly. JDBC is a Java API providing a framework for dealing with tabular and, generally, relational data. Using it in conjunction with Java provides a portable solution to writing distributed database applications.

AdventNet Agent Toolkit

The AdventNet Agent Toolkit is a development tool used for building agents based on the JMX with support for multiple protocols including SNMP, HTML and CMIP [8]. In addition to JMX it also supports building cross platform Java based standalone SNMP agents.

The agent generation process using the AdventNet toolkit is similar in concept to the JDMK implementation since both toolkits are based on JMX technology. Tools are provided for the compilation of MIBs and the generation of skeleton code. Database functionality is then added to instrument the generated agent.

DynamicTMN is a Java toolkit based on the TMN/C++ API standard for CMIP agent and manager application development [9]. The DynamicGDMO Agent Java API provides a flexible GDMO level framework for agent development. The agent API implements a high level Managed Object representation complete with attribute storage and default CMIS operation handlers (see Figure 6).

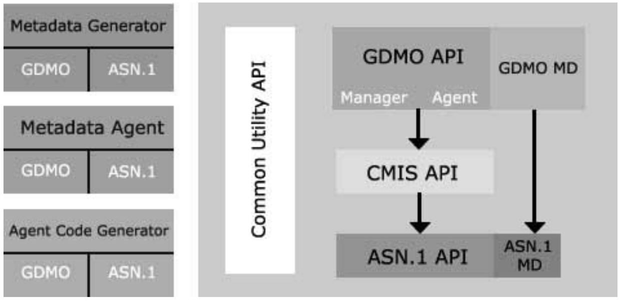


Figure 6. DynamicTMN Toolkit Architecture [11]

Figure 7 outlines the tools and technologies working in unison in the agent – manager configuration.

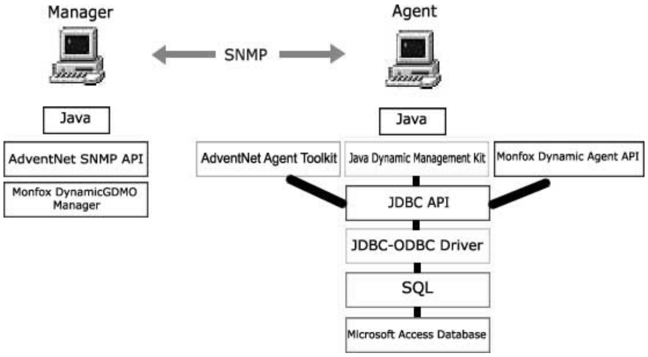


Figure 7. Agent Development Tools and Technologies

3.2 Java SNMP and CMIP Manager classes

The WBEM system uses provider classes to obtain dynamic data from managed resources. These providers incorporate the SNMP and CMIP functionality needed to access agents, requiring the development of Java classes to perform SNMP and CMIP operations on the agent. The manager classes are developed in Java because of its platform independence and because the SNIA CIMOM is developed using Java technology. Application

Programming Interfaces (API) are required to add management protocol functionality to Java classes.

The AdventNet SNMP API provides a development environment for building cross platform, Java and Web based management applications and applets [10]. It can be used to build system management, application management and network management applications and applets. The DynamicTMN GDMO Manager Java API implements GDMO-level classes for manager application development. This high-level API provides the user with the components required to easily implement complex CMIP manager applications without dealing with low-level CMIS messaging.

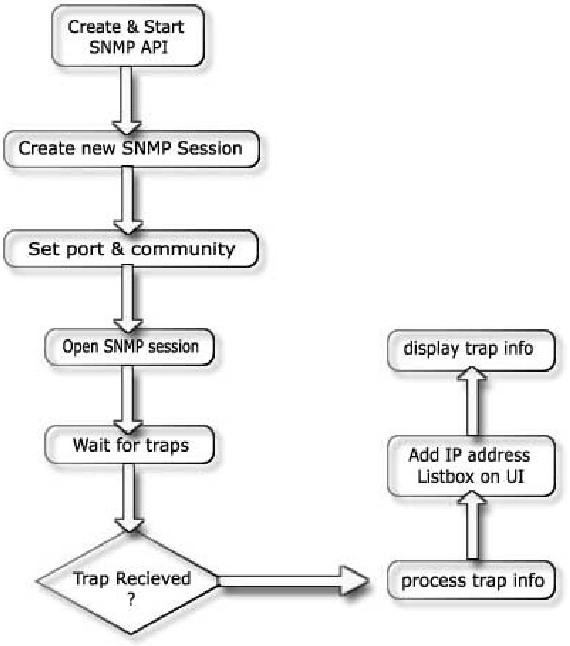


Figure 8. SNMP manager class

The steps involved in developing an SNMP manager class are outlined in Figure 8. Development of the CMIP manager class was similar in concept to that of the SNMP manager class. These manager classes are then inherited by the provider classes located in the WBEM server.

4. A Web Based Manager

To provide a truly platform independent network manager it was decided to develop a web based heterogeneous network manager as opposed to a visual Java application. The web based approach, incorporating technologies such as Java Server Pages (JSP) and Javabeans, has many advantages over a Java application. Firstly it allows the network to be managed from any platform running a web-browser. Secondly, the resource requirements are much less than that of a Java manager, so managing the network from a handheld device is also a possibility. To improve performance it was decided to use JSP and Javabeans as opposed to Java applets. JSP is a technology for developing web pages that include dynamic content [10]. Unlike a plain HTML page, which contains static content that always remains the same, a JSP page can change its content based on any number of variable items, including the identity of the user, the user's browser type, information provided by the user, and selections made by the user. A JSP page contains standard markup language elements, such as HTML tags, just like a regular web page. However, a JSP page also contains special JSP elements that allow a JSP enabled server to insert dynamic content in the page. When the user asks for a JSP page, the server executes the JSP elements, merges the results with the static parts of the page, and sends the dynamically composed page back to the browser, as illustrated in Figure 9.

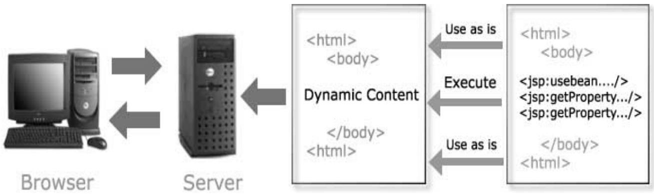


Figure 9. Generating Dynamic Content with JSP

A Javabeans component is a Java class that follows certain coding conventions, so it can be used by tools as a component in a larger application. A bean is often used in JSP as a container for the dynamic content to be displayed by a web page. Typically a bean represents something specific, such as a person, a product, or a shopping cart. A bean is always created by a server process and given to the JSP page. The page then uses JSP elements to insert the bean's data into the HTML template text.

4.1 Developing the Manager Application

As previously mentioned, the web application is designed using JSP and Javabeans. Using Javabeans in conjunction with JSP allows for the building of an application that's easy to maintain and extend as requirements change. The application is based around two Javabeans. These are responsible for user validation operations on the CIMOM. The sequence diagram as illustrated in Figure 10 outlines the steps involved in connecting to the CIMOM. Following the establishment of a connection to the CIMOM the user is presented with the Management Console page (see Figure 11). The page consists of an image map created using Fireworks. The image map provides the user with a graphical view of the simulated heterogeneous network. Clicking on a managed resource loads the configuration page for that resource.

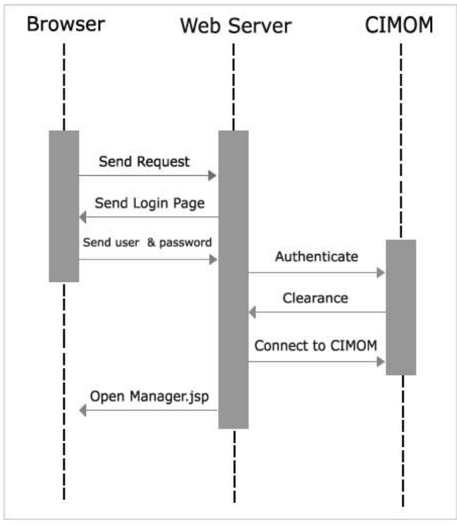
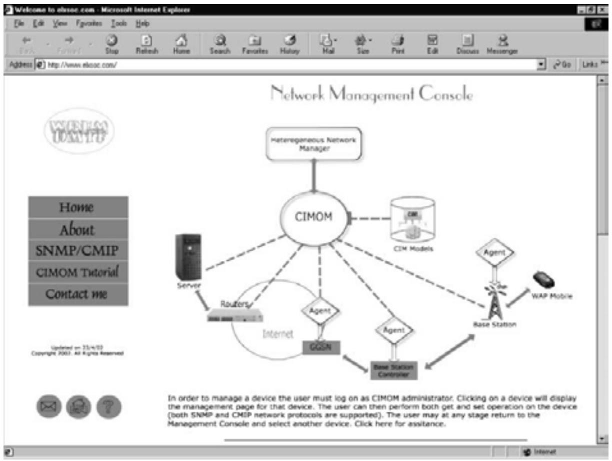


Figure 10. Web Application Sequence Diagram

Figure 12 shows the SNMP server configuration page. All MIB values are retrieved on initial loading of the page. The user can then select which variables they wish to operate on and set a new value. Clicking a submit button causes the page to reload and a Javabean method performs the necessary management operations on the agent, an SNMP SET operation in this instance.

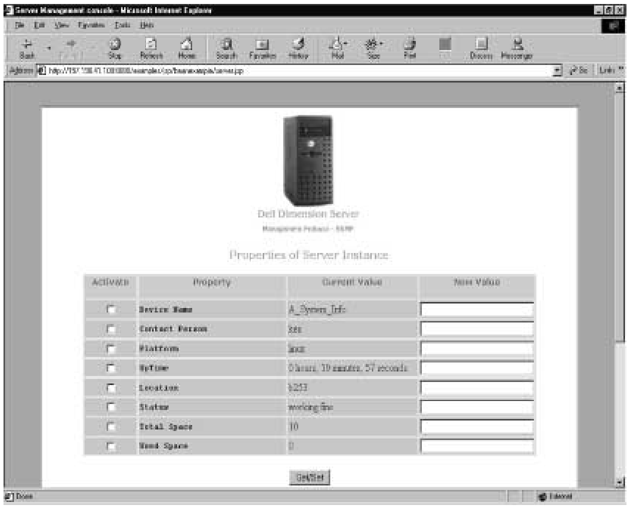
Future work will include the generation, detection and logging of SNMP and CMIP traps by the CIMOM. This will be achieved through the use of the CIM event model. The CIM event model is a framework for handling CIM events and conforms to the Common Information Model Indications Specification, published by the DMTF.


Welcome to ebsoc.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print Edit Discuss Messenger
Address http://www.ebsoc.com/
Network Management Console
WBEM DMTF
Home
About
SNMP/CMIP
CIMOM Tutorial
Contact me
Heterogeneous Network Manager
CIMOM
CIM Models
Server
Routers
Internet
Agent
GGSN
Agent
Base Station Controller
Agent
Base Station
WAP Mobile
Internet

Updated on 23/4/03
Copyright 2003. All Rights Reserved

In order to manage a device the user must log on as CIMOM administrator. Clicking on a device will display the management page for that device. The user can then perform both get and set operation on the device (both SNMP and CMIP network protocols are supported). The user may at any stage return to the Management Console and select another device. Click here for assitance.

Figure 11. Web Based Manager


Server Management console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print Edit Discuss Messenger
Done
Internet

Address http://157.190.41.100:8080/examples/jsp/beanexample/server.jsp

Dell Dimension Server

Management Protocol - SNMP

Properties of Server Instance

Activate	Property	Current Value	New Value
□	Device Name	A_System_Info	
□	Contact Person	[illegible]	
□	Platform	linux	
□	UpTime	0 hours, 19 minutes, 57 seconds	
□	Location	b253	
□	Status	working fine	
□	Total Space	10	
□	Used Space	0	

Get/Set

Figure 12. Typical Configuration page

Table 1. Summary of Tools and Technologies

Product Name	Function
Sun JDMK	SNMP Agent development kit
AdventNet Agent Toolkit	SNMP Agent toolkit
AdventNet SNMP API	SNMP interfaces to agents
Monfox DynamicTMN	CMIP Manager/Agent toolkit
Java Server Pages	Generation of dynamic content
Javabeans	Web based management
SNIA CIMOM	WBEM implementation

4. Conclusion

Development of the architecture for the management of a heterogeneous network is presently in its mid stages. The tasks completed to date include:

- Simulation of a heterogeneous network through the construction of Java based SNMP agents and CMIP agents.
- The establishment of the WBEM system
- Development of a web based heterogeneous network manager.

It is also envisaged to incorporate Bluetooth technology into the heterogeneous network. Bluetooth is a specification to enable short-range wireless communications and through the use of the Serial Link over IP protocol it is possible to communicate with SNMP and CMIP agents over the Bluetooth serial link.

This paper has shown how various industry technologies and tools, as outlined in Table 1, can be successfully integrated into the WBEM to develop a platform independent heterogeneous network management system. All tools and technologies integrated seamlessly into the WBEM server architecture and the client side Java technologies such as JSP and Javabeans proved to be an excellent choice for the development of the web-based manger because of their ease of use and platform independent architecture. As users' understanding of the benefits of WBEM and particularly CIM grow, WBEM implementations will increase accordingly. This holds great promise for the future of managing heterogeneous networks.

Acknowledgements

The authors acknowledge the support of Monfox in the use of DynamicTMN for the work reported in this paper.

References

[1] Winston Bumpus, Andrea Westerinen – Common Information Model, Wiley, Oct 2000
[2] JavaOne 2001, Jim Davies, Andrea Westerinen, Guru Bhat - Developing WBEM Applications and Instrumentation for the Java(TM) Platform
[3] Distributed Management Task Force, Inc. (July 20th, 1999). Specification for the representation of CIM in XML v2.0, URL: http://www.dmtf.org/download/specs/xml
[4] Sun WBEM SDK - http://wbemservices.sourceforge.net/
[5] SNIA CIMOM – http://www.snia.org
[6] William Stallings – SNMP, SNMPv2, and CMIP, Addison-Wesley, Oct 1994
[7] Java Dynamic Management Kit - http://sun.java.com/jdmk4.2
[8] AdventNet SNMP API - http://www.adventnet.com/products/snmp/index.html
[9] Monfox DynamicTMN toolkit - http://www.monfox.com/protocol.htm
[10] Hans Bergsten – Java Server Pages, O’Reilly, Jan 2001