Title of IM2009 Application Session Paper

Author(s)

Affiliation(s)

Email: {john, marry}@xyz.edu

Abstract

The abstract is usually limited to 200 words and should concisely state (i) what was done, (ii) how it was done, (iii) principal results, and (iv) their significance.

Abstract should not contain the lengthy back ground of this study, the detail explanation of basic technologies, and the related works.

Introduction

- Web-based Network Management is the use of this technology to manage networks and systems
- Key technologies
  - HTML, HTTP, Web Browser & Servers, Java, CGI, XML, etc.
- Industry Standards for Web-based Network Management
  - Web-Based Enterprise Management (WBEM from DMTF)
  - Java Management eXtension (JMX from Sun)
- Benefits of Web-based Network Management
  - Reduced development costs by using open technology
  - Unification for separated management platforms
  - Simplification by ubiquitous and standard user interface

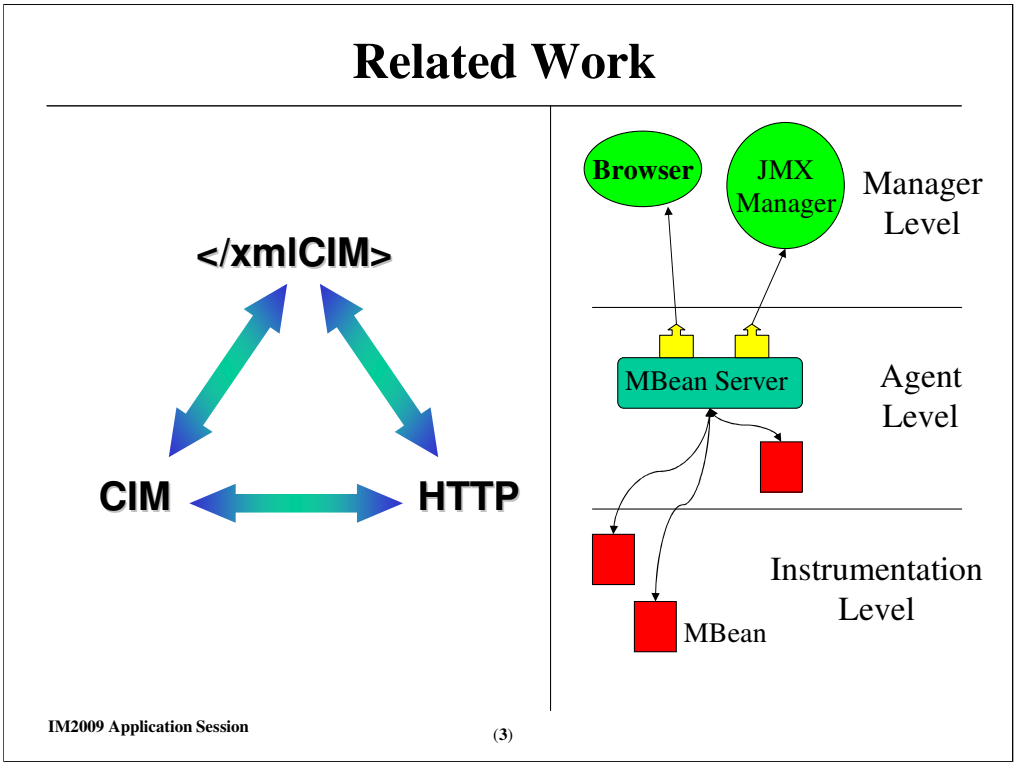
1. Introduction

Following items must be included as separate paragraphs in the introduction:

- Motivation / Background of the study : why this research area is important & valuable ?
- Brief problem specification : why the problem to be solved is important ?
- Related works : the approaches or schemes proposed by other papers, major research results of other papers & studies, standardizations of related works, and their *limitations or shortcomings.*
- Contribution(s) of this paper: Main approach and contribution(s) of this paper to *the problem*
- Brief comparisons with other researches in approach, method and result : what is the difference of this approach from the other's, what are the improvements (in brief)

Following items must **not** be included in introduction:

- narrative explanation of general issues which is related to the area, but not specific to the problem that this paper is try to solve



2. Related work

Recently, there are two promising approaches in Web-based management from industrial standardization bodies: Web-Based Enterprise Management (WBEM) [8] and Java Management eXtension (JMX) [9]. WBEM multi-vendor alliance launched in July 1996 and worked for establishing standards for Web-based network management software. In 1997, WBEM adopted HTTP as its transfer protocol and selected the Extensible Markup Language (XML) [14] as a representation for management information.

DMTF and WBEM worked together by giving the way for the encoding of the Common Information Model (CIM) schema in XML [14, 15]. The CIM is an object-oriented information model, standardized within the DMTF for the purpose of providing a conceptual framework within which any management data may be modeled. Allowing CIM information to be represented in the form of XML brings all the benefits of XML and its related technologies to management information which uses the CIM meta-model [15]. The XML encoding specification defines XML elements, written in Document Type Definition (DTD), which is used to represent CIM classes and instances. The encoded XML message could be encapsulated within HTTP. Further, WBEM defines a mapping of CIM operations onto HTTP that allows implementations of CIM to operate in a standardized manner. Much work in WBEM is currently under way: seventeen working groups are updating specifications. The result from WBEM is fairly stable, but still not quite ready for deployment.

Another promising approach to the Web-based management is being realized by Sun: JMX (formally JMAPI) [9, 16]. Sun announced JMX in order to provide ubiquitous management framework and promote an abundance of management application in Java. Based on the early JMAPI work as well as research taken from Java DMK development, JMX ended public review in July 1999 and is awaiting completion of the reference implementation [16].

Architecture for EWS-based NM

- Extended version of EWS-based element management architecture: POS-EWS
- Thin-Client & Fat-Server paradigm
- Two model: 2-tier and 3-tier architectures
- Uses Java technology
- Communications using HTTP
- Operation & Information Encoding
- Supported basic management functions
  - Notification, Data collection, Agent discovery, Data setting

IM2009 Application Session (4)

3. Design concept

3.1 Architecture for EWS-based NM

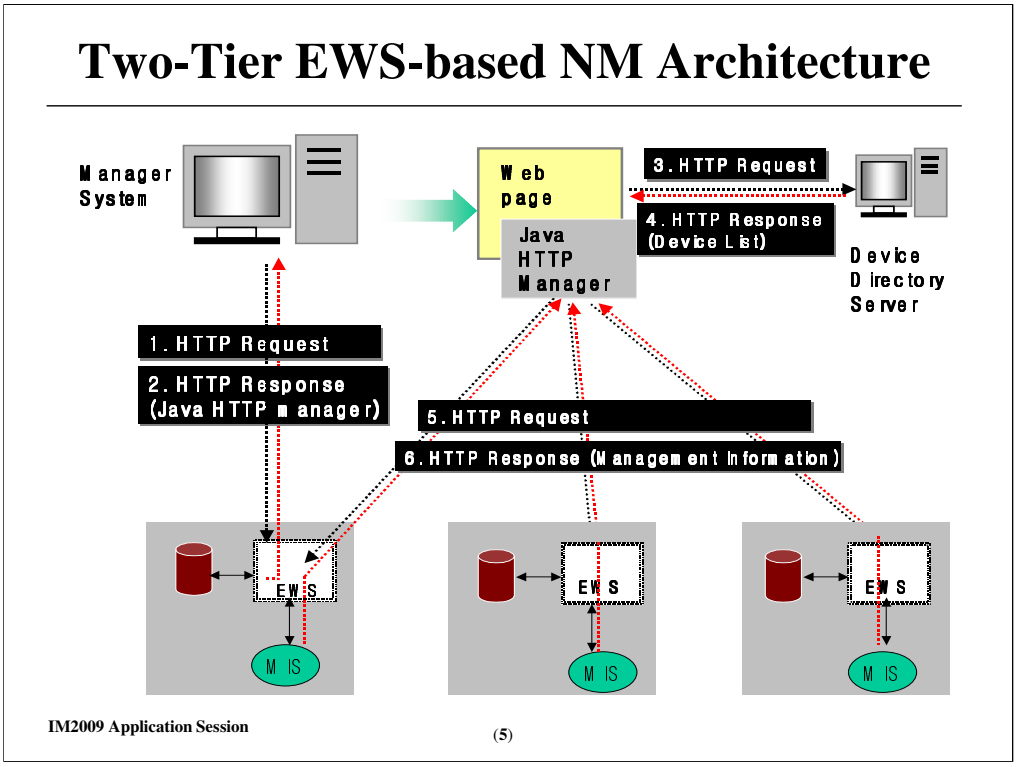
With the defined target domain, we designed the architecture for an EWS-based network management system. In the previous work, we proposed an element management architecture having an EWS as a core component. We have developed an HTTP/1.1 compliant embedded Web server (called POS-EWS) that supports our proposed architecture [4, 13]. We also extended the EWS-based element management architecture to the network management architecture.

We applied the thin-client and fat-server paradigm to the architecture. From numerous computing resources in the device, we deduced this design concept. The first extended version from element management architecture is 2-tier architecture, thereafter modified to a 3-tier architecture as a provision for the lack computing resources in device.

We use Java technology, especially the Java applet. The Java applet is downloaded from the Web server and run on a Web browser. The Java applet is a mobile code over the Internet. They are stored in Web server and executed on the browser. There is no Java execution environment on devices. The Java applet has an inherent security problem: it is restricted in accessing local disk, executing another program and network connection to other hosts by Java applet security mode (Sandbox model) [10]. Code signing extends applet capabilities to make network connection with other hosts.

HTTP is used as transport protocol between EWS-based agent and management station. We defined an information encoding scheme for management data. The encoded message is encapsulated into HTTP message. With the encoding scheme, management operation such as get and set are encoded into URL that is part of HTTP message [5, 11].

Our EWS-based network management system supports four basic management functions: notification, data collection, agent discovery and data setting. Notification determines which events in device occurs on the basis of the event message and customizes the event message to notify administrator. Data collection gathers management data from the device and stores the data it collects in the database. It also performs threshold monitoring and generates threshold events. Agent discovery polls an EWS agent to initially discovery EWS equipped device and then detects configuration and status changes in the network. Data setting provides an administrator a mechanism for changing management information of a device.



3.2 Two-tier EWS-based NM architecture

HTTP is a client-server scheme. One of its side effects is that first Web page is served from a Web server to a Web browser, and subsequent Web pages cannot be included in the Web display except for image. For a network management system, this is very strict limitation. Network management system must gather management information from multiple devices, and formatted management data from multiple devices can be placed in the same page. This is why Java applets are used [10, 12].

Java applets are downloaded by a browser. Once the applet is loaded, it can control the location from where it receives its data and how to display or manipulate that data. Java applets by nature are cross-platform and act the same within any browser. Fortunately, it is a straightforward task to design an applet to make connections with multiple devices if Java applets are programmed on the basis of a Java security model and signing utility.

Java implementation of an HTTP manager is a key component in a 2-tier architecture. The Java HTTP manager source code is written and compiled to produce a Java HTTP manager applet. This applet is stored in a network device and is transferred by the EWS to the browser over the network at run time. After loaded on the JVM of a browser, the Java HTTP applet communicates with EWS agents in the network and enables the administrator to control and monitor the network devices using HTTP [5, 10, 12].

An administrator starts the management task by making Web browser connection to one of the managed devices. The connected device responds with a Java HTTP manager applet. First operation of the Java HTTP manager applet is to retrieve registered device list from the Device Directory Server. After retrieving device list, The Java HTTP manager communicates with devices specified in the list to perform a management task. Management Information Server (MIS) responds to the Java HTTP manager request. It performs basic management functions explained before: Notification, Data collection, Agent registration.

Communications using HTTP

- Java HTTP manager and management information server communicate using HTTP
- Avoid new specific management protocol
  - Reuse existing communication protocol
- Avoid overhead of frequently setting up and tearing down connections
  - By persistent TCP connection of HTTP 1.1
- Management operation encoding in URL
- Management information encoding in HTTP

IM2009 Application Session (6)

3.3 Communication using HTTP

Java HTTP manager and management information server communicate using HTTP. By reusing existing communication protocol, developers can avoid adding new specific management protocols. HTTP 1.1 compliant Web Server supports persistent TCP connection [5]. Persistent connection allows multiple requests to be pipelined on a single connection, with a mechanism to encode each response as a series of chunks, making it unnecessary to buffer the entire response before transmission. This mechanism avoids overhead of frequently setting up and tearing down connections. HTTP is a TCP-based application protocol, therefore it is more reliable than UDP-based application protocol such as SNMP.

In order to manage network resources using HTTP, a Java HTTP manager can specify management operation with the name of managed resource. We define a mapping between URL and management operation with the name of managed resource. The mapped URL is compliant with the standard URL syntax, therefore it can be handled by a conventional Web server and Web browser [11]. We define three management operations: get, set, getnext. The format of mapped URL syntax is depicted below.

http://host/resource/management-operation?parameter

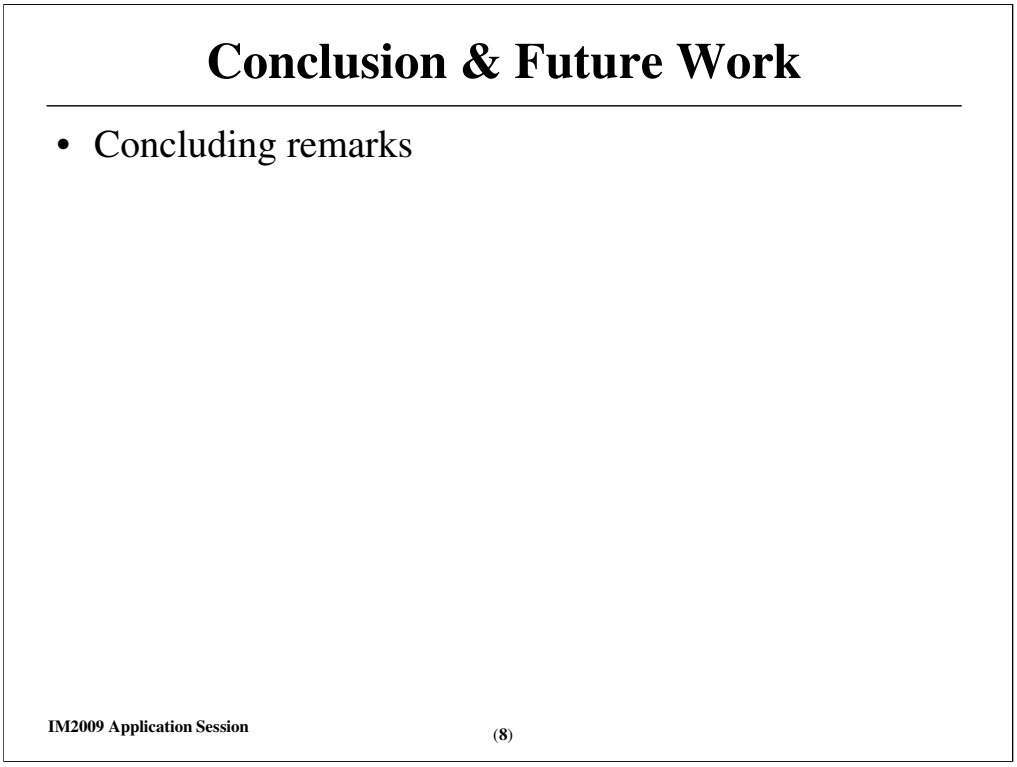
Management information is encoded into the HTTP data part in chunk of binary data of arbitrary size. Java HTTP manager request management operation to the management information server [5]. The management information server responds with an HTTP header followed by a MIME-typed proprietary encoded data. This data can be compressed with gzip if the Web browser supports the decompression capability. The format of TCP payload of HTTP message transferring encoded management information is depicted bellow.

HTTP Header	**MIME Type**	***gzip*'ed data**

Performance Analysis

- Performance analysis 1
  - major results
  - implications
- Performance analysis 2
  - major results
  - implications

4. Performance Analysis



5. Conclusion and Future work

There are lots of benefits in applying Web technology to network management. Most commercial network devices are equipped with EWS, which are used for element management only. We have proposed an EWS-based network management architecture. We assume that the target network is composed of the same functional devices from different vendors or different devices from one vendor.

We have applied the thin-client and fat-server paradigm to the architecture and used Java technology, especially the Java applet. HTTP is used as transport protocol between management entities. On the basis of proposed architecture, we are currently implementing the management system. Our future work involves enhancing our architecture for application to an open (heterogeneous) target network.

References

[1] McCombie, B., "Embedded Web servers now and in the future," Real-Time Magazine, no.1, Mar. 1998, pp. 82-83.

[2] Ian Agranat, "Embedded Web Servers in Network Devices," Communication Systems Design, Mar. 1998, pp. 30-36.

[3] Ching-Wun Tsai and Ruay-Shiung Chang, "SNMP through WWW," International Journal of Network Management, Vol 8. 1998, p. 104-119.

[4] M. J. Choi, et. al., "An Efficient and Lightweight Embedded Web Server for Web-based Network Element Management," IEEE/IFIP Network Operations and Management Symposium, Hawaii, April 2000, pp.187-200.

[5] R. Fielding, et. al., "Hypertext Transfer Protocol - HTTP/1.1," RFC 2616 IETF HTTP WG, June 1999.

[6] D. Raggett, et. al., "HTML 4.01 Specification," IETF HTML WG, http://www.w3.org/TR/html401, Dec. 24 1999.

[7] J. Case, et. al., "The Simple Network Management Protocol (SNMP)," RFC 1157, May 1990.

[8] WBEM, "WBEM Initiative," http://www.dmtf.org/wbem/.

[9] Sun, "Java Management Extensions (JMX)," http://java.sun.com/products/ JavaManagement/, Jun. 1999..

[10] Sun, Java Management Programmer's Guide, Developer's Release, June 1996.