XML-based Configuration Management for Distributed System

Hyoun-Mi Choi, Mi-Jung Choi, James W. Hong
Dept. of Computer Science and Engineering, POSTECH, Korea
Email: {siwa, mjchoi, jwkhong}@postech.ac.kr

Abstract

Today, we are seeing more distributed systems on enterprise networks and throughout the Internet. In general, a distributed system is composed of many subsystems. It is difficult to effectively manage the configuration information of distributed systems because they are distributed in nature and may be deployed with different software components, all of which may run on heterogeneous computing platforms. The configuration information of a subsystem has relations with the information of other subsystems in a distributed system. A centralized configuration management system is necessary for distributed systems. In this paper, we propose the design of X-CONF (Xml-based CONFiguration management system) using XML technologies. X-CONF automatically recognizes changes to configuration information among distributed system components via a formal structure of configuration information specifications. We have designed a flexible and interoperable configuration management system by applying Simple Object Access Protocol (SOAP) as a communication method. Also, X-CONF provides a Web-based user interface to administrators for ubiquitous access. For validation, we have developed an XML-based configuration management system for NG-MON, which is a distributed and real-time Internet traffic monitoring and analysis system.

1. Introduction

Most large-scale software systems are usually implemented with many computers to distribute the processing. In this paper, the components of a distributed system are called subsystems. A configuration management system is needed to efficiently manage the configuration information in each subsystem. The larger the distributed systems are, the more various the relations of the configurations and the more diverse implementation environments among member subsystems become. A centralized configuration management system is required to consider the relationship to maintain consistency of configuration information and to provide a platform and language independent method of communication.

The Simple Network Management Protocol (SNMP) [1] is the most widely used method for network management on the Internet. Only one object value defined by a MIB is accessed through an SNMP Get operation. The GetBulk operation must check the limitation of the message size that can be retrieved in one request. SNMP MIB is based on a simple hierarchical structure of management information, because it is difficult to present dependencies among managed objects. XML [2] can represent the complex structure of management information using any tag. An XML-based protocol has been proposed as an alternative to the SNMP protocol, which complements the constraints mentioned previously.

In this paper, we present the design and implementation of X-CONF, which uses XML technologies to implement a configuration management system for distributed systems. X-CONF first defines the configuration information with the XML Schema [3], which provides a powerful and extensible modeling capability. The management information model must be general and effective in order to be easily applicable to the configuration management of various distributed systems. X-CONF exchanges messages between a manager and agents in subsystems using SOAP [4]. Based on XML and SOAP, the XML-based manager can directly call management operations in the agent via SOAP RPC. Also, SOAP supports a standard description of operations using Web Services Description Language. Therefore, it is very convenient to develop and extend the management operations. SOAP is generally accepted as a sufficient transport method for Netconf [5]. In addition, SOAP is platform-independent and this places no restrictions on endpoint implementation technology choices. SOAP messaging is therefore interoperable with any platform and any device.

2. Related Work

In this section, we introduce related work on XML-based configuration management. Current configuration management with XML is bounded to configure network devices, namely, the element level. It is inappropriate to apply this to the configuration management of distributed systems.

- **IETF Network Configuration**: The Network Configuration (netconf) Working Group was formed in May 2003. The Netconf Working Group is chartered to produce a protocol suitable for network configuration. The Netconf protocol will use XML for data encoding purposes, because XML is a widely deployed standard which is supported by a large number of applications. XML also supports hierarchical data structures. The working group suggests Netconf configuration protocol as a starting point.
- **Cisco's Configuration Registrar**: The Cisco Configuration Registrar is a Web-based system for automatically distributing configuration files to Cisco IOS network devices. The Configuration Registrar works in conjunction with the Cisco Networking Services (CNS) Configuration Agents located at each device. The Configuration Registrar delivers the initial configuration to Cisco devices when starting up on the network for the first time. The Configuration Registrar uses HTTP to communicate with agents, and transfers configuration data in XML. The Configuration Agent in the devices uses its own XML parser to interpret the configuration data from the received configuration files.
- **Juniper Networks' JUNOScript**: Recently, Juniper Networks introduced JUNOScript for their JUNOS network operating system. JUNOScript is part of their XML-based network management effort and uses a simple model, designed to minimize both the implementation costs and the impact on the managed device. JUNOScript allows client applications to access operational and configuration data using XML-RPC. JUNOScript defines the DTDs for the RPC messages exchanged between client applications and JUNOScript servers running on the devices. Client applications can request information by encoding the request with JUNOScript tags in the DTDs and sending it to the JUNOScript server. The JUNOScript server delivers the request to the appropriate software modules within the device, encodes the response with JUNOScript tags, and returns the result to the client application.

3. Requirements

The subsystems have close relations with one another's configuration information. That is, configuration information is shared among the component subsystems and a change to the configuration information of one subsystem affects the other subsystems. To effectively manage the configuration information of each subsystem, the functional requirements of a configuration management system for distributed systems are as follows.

(1) Show, delete, and modify the configuration information in the subsystem selected by an administrator
(2) Add and delete a subsystem that an administrator selects.
(3) Provide a Web-based user interface for a ubiquitous access.
(4) Sufficiently present complex relations among subsystems in the description of configuration information.
(5) Propose a formal structure to describe configuration information of a subsystem. This formal structure of configuration information must be easily extendable to other

subsystems and should allow an XML-based manager to apply XSL and XSLT for the user interface.

(6) Maintain the consistency of the configuration information among remote subsystems. As well, whenever a subsystem is added or deleted, the manager also reflects the new information in the configuration information of the other subsystems related to it. Then, the subsystems must reboot themselves in order to apply the updated configuration information.

(7) Communicate between a manager and an agent in each subsystem regardless of the implementation environment.

4. Design of X-CONF

Figure 1 illustrates the configuration information of each subsystem. An all_info element, a child element of the Configuration element, contains the configuration information shared with all subsystems. Any element can be a child element of all_info element according to the configuration information. A group_info element describes the configuration information shared within group. A subsys element contains the configuration information independent of other subsystems. A manager must know the meaning of this formal structure. By identifying the parent element to which the changed configuration information belongs, the manager decides where to deliver the changed configuration information.

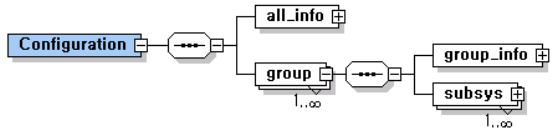

Figure 1. XML Schema of the Configuration Information

Figure 2 illustrates the detailed architecture of X-CONF, in which a centralized XML-based manager controls the configuration information of subsystems equipped with an XML-based configuration agent. A manager is divided into four modules: XMLDB handler, XSL/XSLT processor, SOAP client, and Operation module. The XMLDB handler module retrieves and modifies the configuration information from XMLDB and inserts the new configuration information into the XMLDB. The XML-based configuration management agent contains a SOAP server module, which defines the methods to be called from a manager, an Operation module that contains the applications of a subsystem, and an XML parser module.

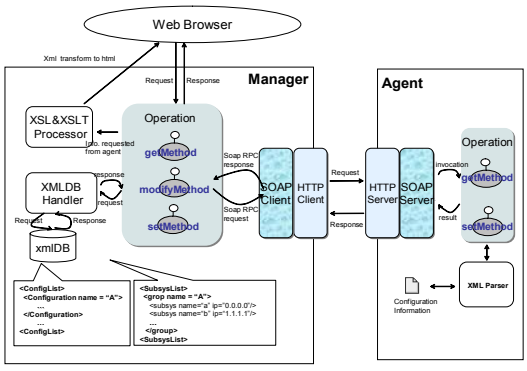

Figure 2. Architecture of X-CONF

5. Implementation

The distributed system applying the architecture of X-CONF is NG-MON [6], which is a passive network monitoring system with a cluster architecture for load distribution. This section explains the X-CONF implementation environment and shows a sample of the configuration information in a subsystem.

We refer to Apache Project Group which provides an Application Program Interface (API) implemented with JAVA to support related XML technologies. X-CONF needs the following APIs: XML Xerces as an XML parser, Xalan to transform an XML document into other forms, Xindice as an XMLDB and AXIS as a SOAP engine to apply SOAP communication method between the manager and the agents. The manager and the subsystems must setup a Web server for HTTP communication. Considering user interface implementation with Java Server Page (JSP), the manager installs TOMCAT, which is provided in Jakarta Apache. Figure 3 is an example of an XML-based formal structure of the configuration information of a subsystem (packet capture) in the NG-MON. The root element (*configuration*) has three attributes: name (*ng-mon*), ip (141.223.11.1), and target (*packetcapture*). The name attribute is the name of the distributed system. The ip attribute is the IP address of the subsystem. The target attribute is the group name. If a manager asks the subsystems in the same group to simultaneously perform the same management action, the manager can simplify the query by using the group name. An instance is when the manager wants all subsystems in the group to reboot.

```
<configuration name= "ng-mon"
        ip="141.223.11.1"
        target ="packetcapture">
  <all_info>
    <admin name ="mount"
     email="mount@postech.ac.kr" />
    <database user="root"
        password="pwd" />
  </all_info>
  <group name="packetcapture">
    <group_info>
      <device name = "eth1" />
    </group_info>
    <subsys ip = "141.223.11.1" />
      <database user="siwa"
            passwd="123" />
    </subsys>
    <subsys ip = "141.223.11.2" />
        <database user="mjchoi"
            passwd="aaa" />
    </subsys>
  </ group >
```

```
< group name="flowgenerator">
  <group_info>
    <time interval= "2" />
  </group_info>
  <subsys ip = "141.223.11.3" />
  <subsys ip = "141.223.11.4" />
</group >
< group  name="flowstore">
  <group_info>
    <p2p file = "p2p.xml" />
  </group_info>
  <subsys ip = "141.223.11.5" />
  <subsys ip = "141.223.11.6" />
</ group >
<group  name="trafficanalyzer">
  <group_info>
    <p2p file = "p2p.xml" />
  </group_in>
  <subsys ip = "141.223.11.7" />
  <subsys ip = "141.223.11.8" />
</group>
< /configuration >
```

Figure 3. Configuration in Subsystem (141.223.82.1)

6. Conclusion and Future Work

In this paper, we proposed the design and implementation of X-CONF, which effectively manages the configuration information using SOAP as the method of communication between the XML-based manager and the XML-based configuration agents. X-CONF automatically advertises modified configuration information to the related subsystems via the defined XML form when the configuration information shared with other subsystems is modified. This maintains the consistency of the configuration information among the subsystems. We applied X-CONF as the configuration management system for NG-MON. For future work, we will validate the flexibility and extendibility of X-CONF by developing a configuration management system for other distributed systems. Finally, we will propose a general and formal XML Schema for the configuration information specification that will allow it to be easily applied to the configuration management of other distributed systems.

6. References

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

[2] W3C, "Extensible Markup Language (XML) 1.0", W3C Recommendation, October 2000.

[3] W3C, "XML Schema Part0: Primer", W3C Recommendation, May 2001.

[4] W3C, "SOAP Version 1.2 Part 2: Adjuncts", W3C Working Draft, December 2001.

[5] IETF, "Network Configuration (Netconf)", http://www.ietf.org/html.charters/netconf-charter.html.

[6] Se-Hee Han, Myung-Sup Kim, Hong-Taek Ju and James W. Hong, "The Architecture of NG-MON: A Passive Network Monitoring System", 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2002), Montreal, October 2002, pp. 16-27.