Design and Implementation of Manageability Services for Common Management Model

C. Eric Wu

IBM T. J. Watson Research Center
cwu@us.ibm.com
http://alphaworks.ibm.com/tech/gems
Outline

• Introduction
• Persistent Manageability Services
  – Linux Heart Beat Service
  – Linux OS Service
• Manageability Port Types
  – Identity: Managed Resource Factory port type
  – Relationship: Association port type
• Manageability Service Factories and Services
  – Linux Process Factory and Services
  – Disk Partition Factory and Services
  – System Statistics Factory and Services
  – Manageability Service Factory and Services for System Services
• Summary
Goals and Background

• Goals for the development of Manageability Services
  – Resource Management
  – Foundation for Autonomic Computing

• Open Standards for Management
  – Simple Network Management Protocol (SNMP), Internet Engineering Task Force (IETF), 1988
    • Managing TCP/IP networks
  – Web-Based Enterprise Management (WBEM), including Common Information Model (CIM), promoted by Distributed Management Task Force (DMTF) since 1996
    • SNIA’s Java-based CIMOM, Sun’s Open WBEM
    • The Open Group’s C++ based Pegasus, SCO’s CIMOM
    • Microsoft’s Windows Management Instrumentation (WMI)
  – Java Management Extensions (JMX), Sun Micro., 1999
Advantages of Grid Services

• Self-describing and use WSDL to describe service operations and service data elements (SDEs)
  – Clients can learn from WSDL descriptions before invocation
  – Eliminate potential problems resulting from API changes

• SOAP over HTTP
  – Text-based XML document, eliminating byte-ordering problems
  – Allowed to go through corporate firewalls

• Grid services
  – Factories for on-demand services
  – Notification mechanism for information exchange
  – Registry

• Common Information Model (CIM) for Resources
  – Resource properties --- SDEs
  – Control methods --- service operations
CMM and Manageability Services

- **Common Management Model (CMM)**
  - GGF Working Group
  - Builds upon existing resource model standards such as CIM
- **CMM Scope**
  - Manageability of entities represented as Grid services
  - A collection of manageability port types, including lifecycle state, identity, and relationship
- **CMM Potential Implementations**
  - Grid services
  - Grid services + CIMOM with CIMOM providers,
  - Grid services + RMC, Grid services + SNMP…
- **Manageability Services**
  - Based on Globus Toolkit version 3, available at IBM AlphaWorks
  - Native Grid services for managing resources
  - Minimum requirement for prototype implementation
  - Identity port type and relationship port type
CMM Potential Implementations

Grid Service Client

Grid Middleware

Registry

Service Instances

client

Service Instance

CIMOM

Managed Object
Sensors and Effectors

• Basic components for resource management and autonomic computing
  – Sensors: monitors collect management data
  – Effectors: control knobs that are used to change resource configuration, behavior, and allocation
Linux Heart Beat Service

• A persistent service
  – monitors system loads periodically
  – System load = length of ready queue in OS scheduler

• Service operation: float[] getLoadAverages();

• Service Data Element OSLoadAverages
  – Service URL: String service;
  – Timestamp: Calendar timestamp;
  – Load average in past 1, 5, and 15 minutes

• Sleep time is a deployment parameter
  – <parameter name="sleepTime" value="60000"/>

• Act as a notification source
  – Extends Notification Source provider
  – Implements Association port type
GWSDL for Heart Beat Service

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HeartBeat"
    targetNamespace="http://ogsa.globus.org/linux/HeartBeat"
    xmlns:tns="http://ogsa.globus.org/linux/HeartBeat"
    xmlns:hbase="http://ogsa.globus.org/linux/HBbase"
    xmlns:asso="http://ogsa.globus.org/linux/Association"
    xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
    xmlns:gwsdl="http://www.gridforum.org/namespaces/2003/03/gridWSDLExtensions"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
   
   <import location="../../ogsi/ogsi.gwsdl" namespace= "http://www.gridforum.org/namespaces/2003/03/OGSI"/>
   <import location=" ../HBbase/HBbase_port_type.gwsdl" namespace="http://ogsa.globus.org/linux/HBbase"/>
   <import location=" ../Association/Association_port_type.gwsdl" namespace="http://ogsa.globus.org/linux/Association"/>

   <gwsdl:portType name="HeartBeatPortType"
    extends="hbase:HBbasePortType asso:AssociationPortType ogsi:NotificationSource"/>
</definitions>
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HBbase"
targetNamespace="http://ogsa.globus.org/linux/HBbase"
xmlns:tns="http://ogsa.globus.org/linux/HBbase"
xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
xmlns:gwsdl="http://www.gridforum.org/namespaces/2003/03/gridWSDLExtensions"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sd="http://www.gridforum.org/namespaces/2003/03/serviceData"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<import location="../ogsi/ogsi.gwsdl"
  namespace="http://www.gridforum.org/namespaces/2003/03/OGSI"/>
</definitions>
<types>
  <xsd:schema targetNamespace="http://ogsa.globus.org/linux/HBbase"
              attributeFormDefault="qualified" elementFormDefault="qualified"
              xmlns="http://www.w3.org/2001/XMLSchema">
    <xsd:complexType name="OSLoadAveragesType">
      <xsd:sequence>
        <xsd:element name="service" type="xsd:string"/>
        <xsd:element name="timestamp" type="xsd:dateTime"/>
        <xsd:element name="load1" type="xsd:float"/>
        <xsd:element name="load5" type="xsd:float"/>
        <xsd:element name="load15" type="xsd:float"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="EmptyType"/>
    <xsd:complexType name="ArrayOf_float">
      <xsd:complexContent>
        <xsd:restriction base="soapenc:Array">
          <xsd:attribute ref="soapenc:arrayType" arrayType="xsd:float[]"/>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="getLoadAverages" type="tns:EmptyType"/>
    <xsd:element name="getLoadAveragesResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="loads" type="tns:ArrayOf_float"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</types>
Messages, Port Type, SDE

<message name="GetLoadAveragesInputMessage">
   <part name="parameters" element="tns:getLoadAverages"/>
</message>

<message name="GetLoadAveragesOutputMessage">
   <part name="parameters" element="tns:getLoadAveragesResponse"/>
</message>

<gwsdl:portType name="HBbasePortType" extends="ogsi:GridService">
   <operation name="getLoadAverages">
      <input message="tns:GetLoadAveragesInputMessage"/>
      <output message="tns:GetLoadAveragesOutputMessage"/>
   </operation>
   <sd:serviceData name="OSLoadAverages" type="tns:OSLoadAveragesType" minOccurs="1" maxOccurs="1" mutability="mutable" modifiable="false" nillable="false">
      <documentation>OSLoadAverages SDE</documentation>
   </sd:serviceData>
</gwsdl:portType>
</definitions>
Linux OS Service w/Sensors and Effectors

• Linux OS Service
  – Linux OS base port type operations
    • getLoadAverages();
    • executeCommand(String[] cmdArgs);
    • shutdown()
    • reboot()
  – Extends Association port type, Grid Service port type, notification source port type, and OS Parameters port type

• OS Parameters Port Type
  – setParameter() and getParameter() for many OS parameters
  – Utility autowsdl to scan available OS parameters during build

• Port type panels are developed
  – Use Globus Toolkit service browser to access these services
Snapshot of Linux OS Service

### Services
- Services
- WSDL
- Service Data

#### OSParameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>threads-max</td>
<td></td>
</tr>
<tr>
<td>pid_max</td>
<td></td>
</tr>
<tr>
<td>cad_pid</td>
<td></td>
</tr>
<tr>
<td>sysrq</td>
<td></td>
</tr>
<tr>
<td>sem</td>
<td></td>
</tr>
<tr>
<td>msgmnb</td>
<td></td>
</tr>
</tbody>
</table>

#### Load Averages
- 0.18
- 0.11

#### Directory List

```
-rw-r--r-- 1 root root 1092135 Feb 3 17:56 wsant.tar.gz
-rw-r--r-- 1 root root 4096 Jan 31 15:40 xerces-1.4.4
-rw-r--r-- 1 root root 4096 Jan 27 14:17 xerces-2.3.0
```
Identity: Managed Resource Factory Port Type

• Port type operations
  – Object[] enumerateIDs();
    • Returns all resource ids in the factory along with descriptions
  – boolean verifyID(String id);
• SDE EnumeratedIDs
• Port type inheritance in Disk Partition Factory

![Diagram showing class inheritance and methods](image-url)
Relationship: Association Port Type

- Port type operation
  - Object[] enumerateAssociatedIDs(String resource)
  - Given pre-defined resource types such as LinuxOS, LinuxProcess, it returns associated resource URL (GSH)
- Port type inheritance in Linux Process service
Linux Process Factory and Services

- Potential problem w/Grid services + CIMOM + providers
  - Enumerates all managed object instances before selecting one because clients do not know key properties and values
  - Not on demand, wasting system resources

- Linux Process Factory
  - Enumerates resource ids (process ids) along with commands through MRFFactory port type operation enumerateIDs()
    - Client then uses Factory port type operation createService() with a specific id to create a service instance for that process
  - Also extends Grid Service and Notification Source port types

- Linux Process Service
  - Service operation terminate(), and setParameter() for nice value, max numbers of child processes, open files, and stack size
    - A kernel module is developed to access task structure and loaded automatically for modifying process parameters
  - Extends Grid Service, Notification Source, Association port types
  - SDE ProcessState
Snapshot of Linux Process Factory

<table>
<thead>
<tr>
<th>pid</th>
<th>Process Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>11583</td>
<td>bash</td>
</tr>
<tr>
<td>11610</td>
<td>kdeinit</td>
</tr>
<tr>
<td>11611</td>
<td>bash</td>
</tr>
<tr>
<td>17991</td>
<td>mozilla-bin</td>
</tr>
<tr>
<td>20682</td>
<td>cupsd</td>
</tr>
<tr>
<td>32595</td>
<td>ant</td>
</tr>
<tr>
<td>32597</td>
<td>java</td>
</tr>
<tr>
<td>32606</td>
<td>java</td>
</tr>
<tr>
<td>1256</td>
<td>ant</td>
</tr>
<tr>
<td>1258</td>
<td>java</td>
</tr>
<tr>
<td>1270</td>
<td>java</td>
</tr>
<tr>
<td>1573</td>
<td>kdeinit</td>
</tr>
<tr>
<td>1574</td>
<td>bash</td>
</tr>
<tr>
<td>2491</td>
<td>ps</td>
</tr>
</tbody>
</table>

### Factory

Instance name: pid17991

Create Instance

### Grid Service

Namespace: Name: DDescriptions

Timeout:

XPath Expression: XPath Namespace Mappings:
Snapshot of a Linux Process Service
Disk Partition Factory

- Disk Partition Factory port type operations
  - Service operations listPartitions(), createPartition(), and removePartition()
  - Extends Grid Service, Factory, Notification Source, and MRFactory port types

- SDE Disks
  - An array of information items, one for each disk to specify the device, number of cylinders, disk size, etc.

- SDE PartitionInformation
  - An XML expression equivalent to the output of listPartitions()
  - An array of items, one for each disk partition to specify the device, start cylinder, end cylinder, file system id, file system name, etc.
Snapshot of Disk Partition Factory
Disk Partition Services

- Port type operations
  - mount() and umount()
  - makeLinuxFileSystem()
    - Name of file system, i.e. ext2 or ext3
    - Category, i.e. 4KB, 1MB, or 4MB block size each inode represents
    - Optional label
  - pvCreateForLVM() to initialize the partition for use with LVM
    - System id set to 0x8e for Linux LVM

- SDE DiskPartitionState
  - Includes partition size in KBs, flag showing if it is mounted, start and end partition location in sectors

- SDE MountInformation
  - Valid when the partition is mounted
  - Includes mount point, file system, file system size, used size, available size, used percentage, and label
Snapshot of a Disk Partition Service

- DiskPartition:
  - mountDir: OK
  - FileSystem: ext3
  - Type: news
  - Label/Result:
  - makeLinuxFileSystem
  - pvCreateForLVM

- Grid Service:
  - Namespace:
  - Name: DiskPartitionState
  - Timeout:
  - XPath Expression:
  - XPath Namespace Mappings:

xml:ns1:value xmlns:ns2="http://ogsa.globus.org/linux/DiskPartition" xsi:type="ns2:DiskPartitionStateType"
  - ns2:disk
  - ns2:device
  - ns2:partitionNumber
  - ns2:sizeKB
  - ns2:startSector
  - ns2:endSector
  - ns2:numberOfSectors
System Statistics Factory

• Statistics are resources for monitoring system well-being
  – CPU utilization, paging activity, I/O transfer rate, process creation frequency, I/O transfer rate, paging frequency, interrupt frequency etc.

• Built on System Activity and Reporting (SAR) utility

• Statistics Factory port type operations
  – executeSampling(): an asynchronous operation, inputs include
    • Sampling interval in seconds
    • Number of samples
    • Output file name
  – removeSamplingFile()

• SDE EnumeratedIDs
  – Inherited from the managed resource factory port type
Snapshot of System Statistics Factory
System Statistics Services

• 15 port type operations, each of which updates its corresponding SDE
  – updateCPUUtilization()
  – updateIOTransferRate(), etc.

• Background sampling process may be gathering statistics for
  – CPU utilization, Process creation, I/O transfer rate,
  – Paging frequency, interrupt frequency
  – Network packets received and transmitted
  – Sockets, queue lengths, system loads
  – Memory, memory page, swap space
  – Inode, context switching, swapping

• One update operation and SDE for each item listed above
Snapshot of a System Statistics Service

![System Statistics Service Screenshot]

**SAR (System Activity and Reporting) Statistics**

- **Command:** sar -u
- **Interval:** 1
- **Result:** OK

- **Options:**
  - A
  - B
  - C
  - D
  - I
  - O
  - P
  - R
  - S
  - U
  - V
  - W
  - X
  - Y
  - N
  - DEV
  - EDEV
  - SOCK
  - FULL

**Grid Service**

- **Namespace:**
- **Name:** CPUUtilization
- **Timeout:**

**XPath Expression:**

```xml
ns1:value xmlns:ns2="http://ogsa.globus.org/linux/SystemStatistics" xsi:type="ns2:CPUUtilizationT"
```

**XPath Namespace Mappings:**

```xml
```
Manageability Services for System Services

- System services include sendmail, lpd, vsftpd, crond, …
- System Service Factory is a Factory that extends MRFactory, i.e. with
  - Service operation enumerateIDs()
  - SDE EnumeratedIDs
- System service port type operations
  - startService()
  - stopService()
  - getStatus()
- SDE ServiceState
  - A list of process ids for the service
  - Service status, i.e. OK or stopped
  - A flag indicating if it is started
Snapshot for Sendmail Manageability Service
Summary

• **Grid Enabled Manageability Services**
  – Persistent Heart Beat Service
  – Persistent Linux OS Service
  – Linux Process Factory and Services
  – Disk Partition Factory and Services
  – System Statistics Factory and Services
  – Manageability Service Factory and Services for System Services

  – Heart Beat Listener, Prime Number Searching Service
  – Sametime Instant Messaging Services

• **Future directions**
  – Automatic G/WSDL generation from CIM MOF files
  – Manageability services for Windows
  – More manageability services for Linux
  – Autonomic computing with adaptive management software
  – Grid applications