

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

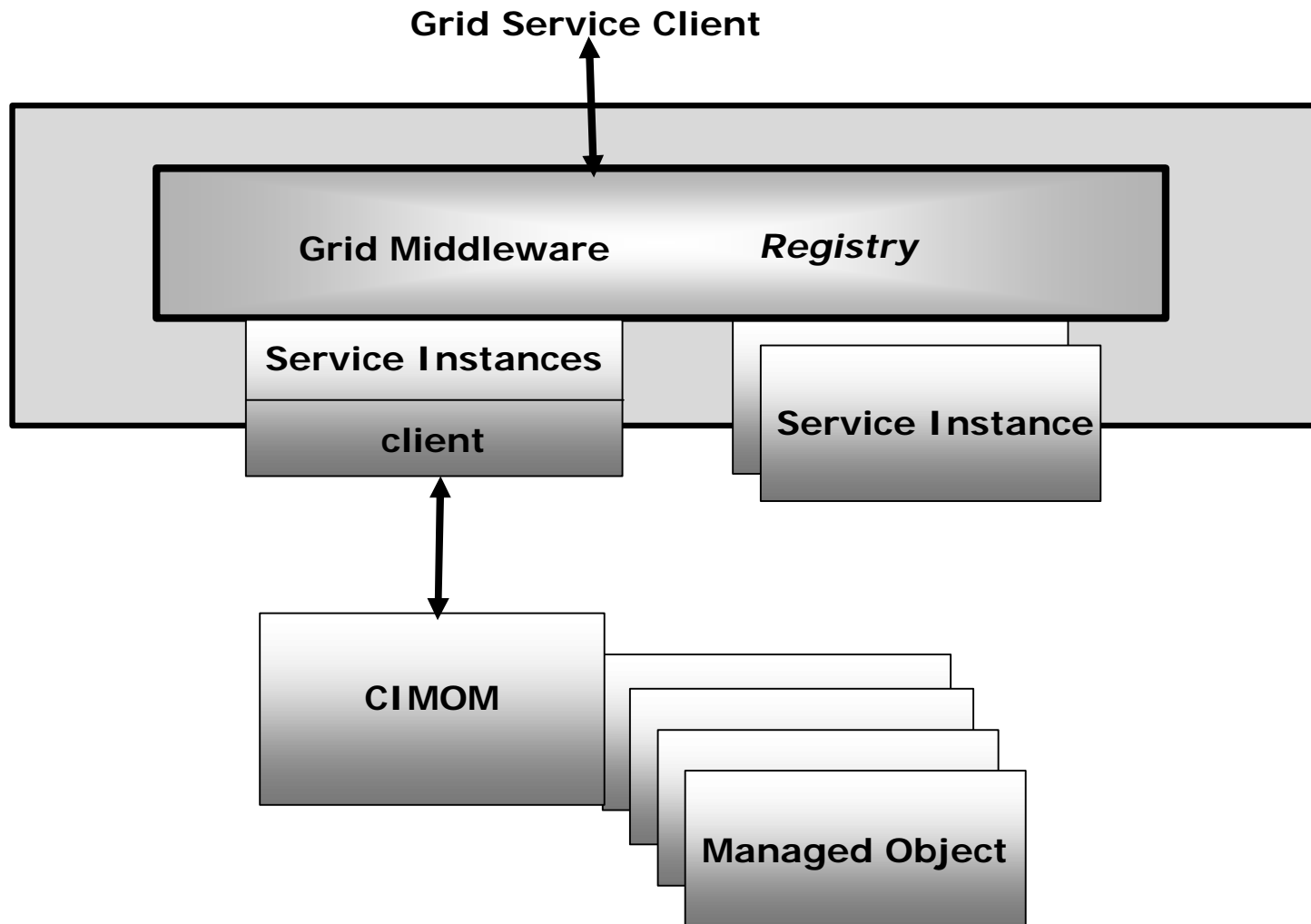
Advantages of Grid Services

- 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

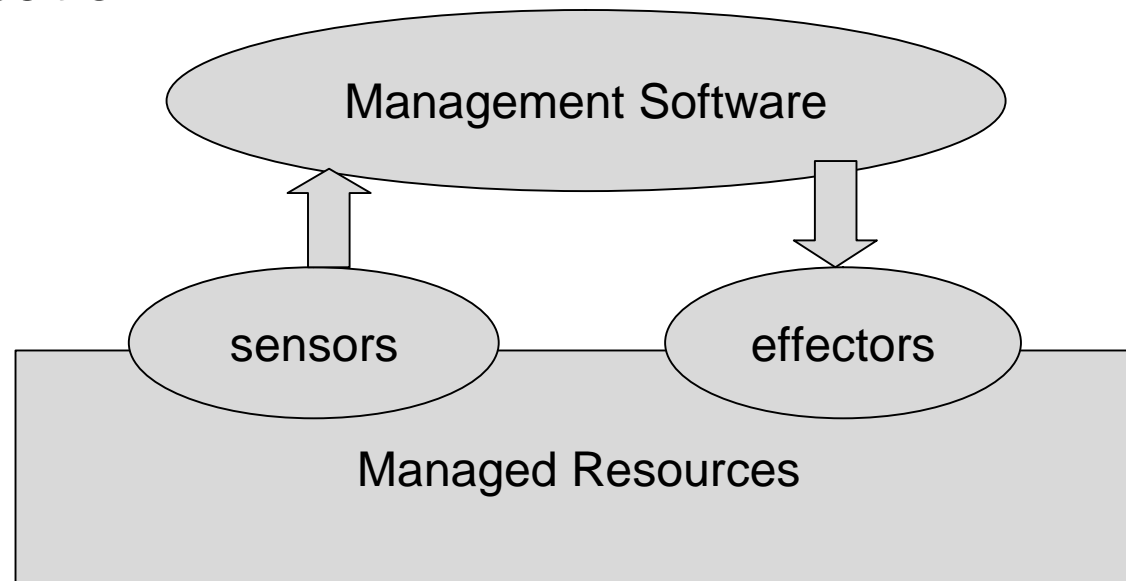
- Common Management Model (CMM)
 - GGF Working Group
 - Builds upon existing resource model standards such as CIM
- CMM Scope
 - Manageability of entities represented as Grid services
 - A collection of manageability port types, including lifecycle state, identity, and relationship
- CMM Potential Implementations
 - 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



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>
```

Namespaces in HBase Port Type GWSDL

```
<?xml version="1.0" encoding="UTF-8"?>  
<definitions name="HBase"  
  targetNamespace="http://ogsa.globus.org/linux/HBase"  
  xmlns:tns="http://ogsa.globus.org/linux/HBase"  
  xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"  
  xmlns:gwsdl="http://www.gridforum.org/namespaces/2003/03/gridWSD  
    LExtensions"  
  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/O  
    GSI"/>
```

Types in HBbase Port Type GWSDL

```
<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

The screenshot shows a web browser window titled "[http://9.2.216.104:8080/ogsa/services/linux/LinuxOSFactoryService/hawk0] - OGSA Service Browser". The address bar contains the URL "http://9.2.216.104:8080/ogsa/services/linux/LinuxOSFactoryService/hawk0" and a "Go" button. Below the address bar are navigation buttons: "Back", "Forward", "New Window", "Close", and "Refresh".

The main content area has three tabs: "Services", "WSDL", and "Service Data". The "Service Data" tab is active, displaying a list of service identifiers: "pid1574", "pid2492", and "pid2616".

Below the list is a section titled "OSParameters". It features a "KernelParameter" dropdown menu set to "threads-max". To its right is another dropdown menu with a list of parameters: "overflowgid", "overflowuid", "pid_max", "threads-max", "cad_pid", "sysrq", "sem", and "msgmnb". The "threads-max" option is currently selected. To the right of this menu is a "setParameter" button.

Below the OSParameters section is a "LinuxOS" section. It displays "Load Averages" with two input fields containing the values "0.18" and "0.11". To the right of these fields is a "Load Averages" button. Below the load averages is an "executeCommand" button.

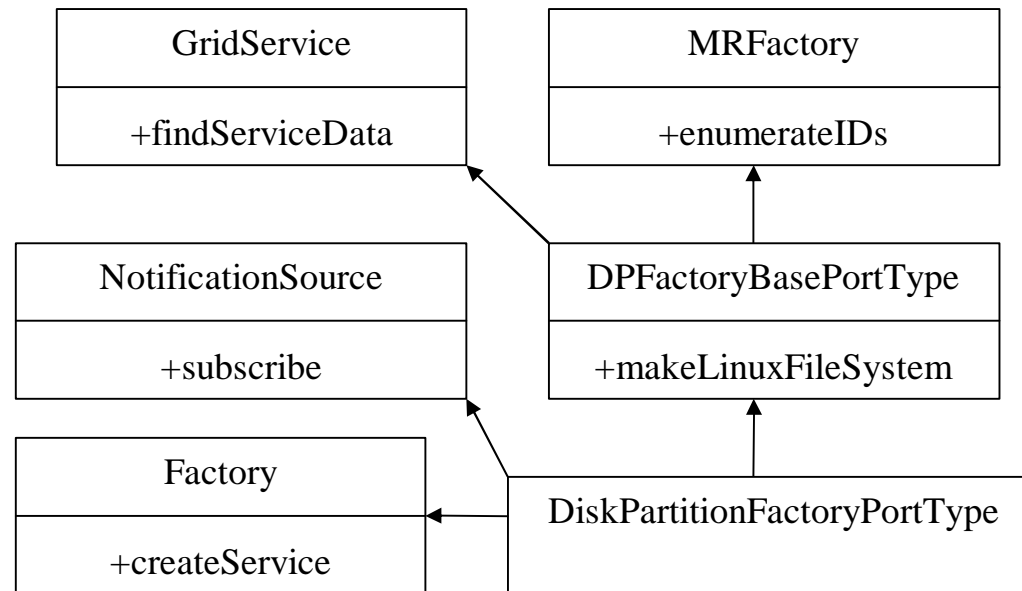
At the bottom of the window is a table listing files and directories. The table has columns for permissions, user/group, size, date, and filename. The data is as follows:

Permissions	User	Group	Size	Date	Filename
drwxr-xr-x	8	cwu	cwu	4096	May 21 17:41 tarz
drwxr-xr-x	5	root	root	4096	May 15 16:55 tmx4j
drwxr-xr-x	13	root	root	4096	Jan 29 13:10 tomcat405
drwx-----	5	501	501	4096	Oct 31 2002 wsad
drwxrwxr-x	10	cwu	cwu	4096	May 27 14:40 wsant
-rw-r--r--	1	root	root	1092135	Feb 3 17:56 wsant.tar.gz
drwxrwxr-x	5	cwu	cwu	4096	Jan 31 15:40 xerces-1_4_4
drwxr-xr-x	5	cwu	cwu	4096	Jan 27 14:17 xerces-2_3_0
drwxr-xr-x	4	cwu	cwu	4096	Nov 17 2002 yhliu

At the bottom of the window are two buttons: "shutdown" and "reboot".

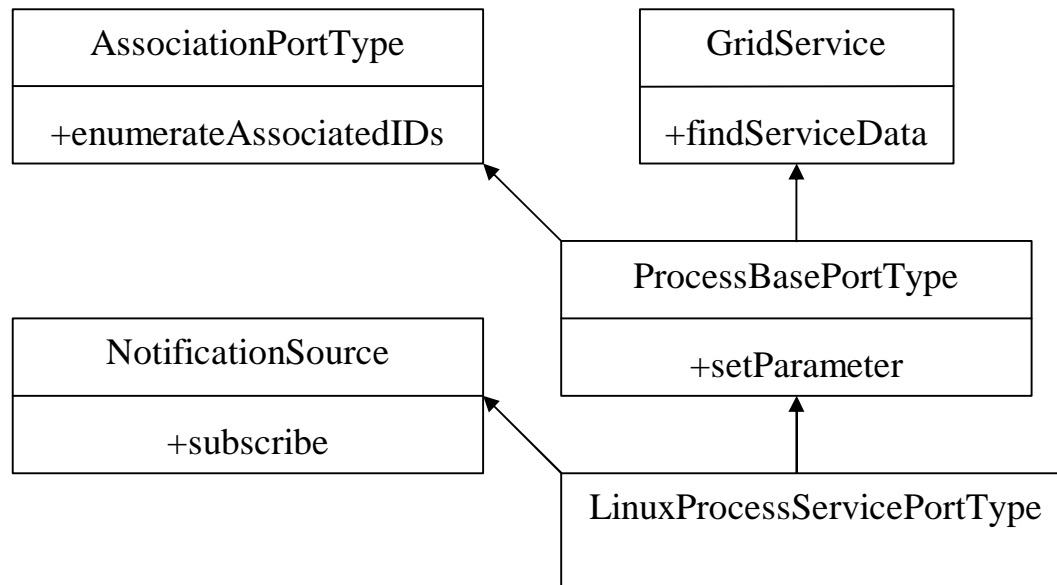
Identity: Managed Resource Factory Port Type

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



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 MRFactory 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

Browser window: [http://127.0.0.1:8080/ogsa/services/base/registry/ContainerRegistryService] - OGSA Service Brows

Navigation: Back Forward New Window Close Refresh

URL: http://9.2.216.104:8080/ogsa/services/linux/LinuxProcessFactoryService Go

Tabs: Services WSDL Service Data

Service: enumerateIDDescriptions

pid11583	bash
pid11610	kdeinit
pid11611	bash
pid17991	mozilla-bin
pid20682	cupsd
pid32595	ant
pid32597	java
pid32606	java
pid1256	ant
pid1258	java
pid1270	java
pid1573	kdeinit
pid1574	bash
pid2491	ps

Factory

Instance name: Create Instance

Grid Service

Namespace: Name: Timeout:

XPath Expression: XPath Namespace Mappings:

Snapshot of a Linux Process Service

The screenshot displays the OGSA Service Browser interface for the service `LinuxProcessFactoryService` at `pid17991`. The browser window title is `[http://127.0.0.1:8080/ogsa/services/base/registry/ContainerRegistryService] - OGSA Service Browser`. The address bar shows `http://9.2.216.104:8080/ogsa/services/linux/LinuxProcessFactoryService/pid17991`. The interface includes navigation buttons (Back, Forward, New Window, Close, Refresh) and a `Go` button. Below the address bar, there are tabs for `Services`, `WSDL`, and `Service Data`. A `terminate` button is visible. The `parameter` section shows `niceValue` with a value of `-10` and a `setParameter` button. The `Grid Service` section includes fields for `Namespace`, `Name` (set to `ProcessState`), and `Timeout`. Below these are fields for `XPath Expression` and `XPath Namespace Mappings`. At the bottom, there are buttons for `Query`, `Subscribe`, and `Unsubscribe`. The main content area displays a list of service parameters:

- `pid xmlns=""`
- `ppid xmlns=""`
- `uid xmlns=""`
- `staticPriority xmlns=""`
- `maxNumberOfChildProcesses xmlns=""`
- `maxNumberOfOpenFiles xmlns=""`
- `maxRealStack xmlns=""`
- `niceValue xmlns=""`

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

OGSA Service Browser

Back Forward New Window Close Refresh

http://9.2.216.104:8080/ogsa/services/linux/DiskPartitionFactoryService Go

Services WSDL Service Data

DiskPartitionFactory

disk: startCylinder: endCylinder: isPrimary: isDosCompatible:

optional partition#:

partitionName:

device	Boot	start	end	#cyls	#blocks	Id	Type
/dev/hdd1	*	1	261	261	2096451	6	FAT 16
/dev/hdd2		262	2116	1855	14900287	131	Linux
/dev/hdd3		0	0	0	0	0	Empty
/dev/hdd4		0	0	0	0	0	Empty
/dev/hda1	*	1	2041	2041	16394301	7	HPFS/NTFS
/dev/hda2		2042	2487	446	3582495	131	Linux
/dev/hda3		2488	2683	196	1574370	130	Linux swap
/dev/hda4		2684	4866	2183	17534947	131	Linux

MRFactory

ID:

hdd1
hdd2
hdd3
hdd4
hda1
hda2
hda3
hda4

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

The screenshot shows a web browser window titled "[http://9.2.216.104:8080/ogsa/services/linux/DiskPartitionFactoryService/hda4] - OGSA Service Browser". The browser's address bar contains the URL "http://9.2.216.104:8080/ogsa/services/linux/DiskPartitionFactoryService/hda4" and a "Go" button. Below the address bar are navigation buttons: "Back", "Forward", "New Window", "Close", and "Refresh".

The main content area has three tabs: "Services", "WSDL", and "Service Data". The "Service Data" tab is active, showing a "DiskPartition" section with the following controls:

- mountDir:** A text input field containing "OK", with "mount" and "umount" buttons to its right.
- FileSystem:** A dropdown menu set to "ext3", a **Type:** dropdown menu set to "news", and a **Label/Result:** text input field, followed by a "makeLinuxFileSystem" button.
- pvCreateForLVM:** A button next to a "Result:" text input field.

Below the "DiskPartition" section is a "Grid Service" section with the following controls:

- Namespace:** A text input field.
- Name:** A text input field containing "DiskPartitionState".
- Timeout:** A text input field.
- XPath Expression:** A text input field.
- XPath Namespace Mappings:** A text input field.

At the bottom of the "Grid Service" section are three buttons: "Query", "Subscribe", and "Unsubscribe".

The bottom of the browser window displays an XML tree view for the query result:

```
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

The screenshot shows a web browser window titled "[http://127.0.0.1:8080/ogsa/services/base/registry/ContainerRegistryService] - OGSA Service Browser". The address bar contains the URL "http://9.2.216.104:8080/ogsa/services/linux/SystemStatisticsFactoryService". The browser interface includes navigation buttons (Back, Forward, New Window, Close, Refresh) and a "Go" button. Below the address bar, there are tabs for "Services", "WSDL", and "Service Data". Under the "Service Data" tab, there are buttons for "Query", "Subscribe", and "Unsubscribe". The main content area displays a list of seven "ns1:value xsi:type='xsd:string'" entries, each with a circular icon to its left. Below this list is a "mystat" label. At the bottom of the window, there is a section titled "Factory of SystemStatistics" with input fields for "fileName" (containing "mystat"), "interval (sec)" (containing "1"), and "count" (containing "100"). Below these fields are buttons for "removeSamplingFile", "executeSampling", and a "result:" label followed by an empty input field.

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

The screenshot shows a web browser window titled "[http://127.0.0.1:8080/ogsa/services/base/registry/ContainerRegistryService] - OGSA Service Browser". The address bar contains the URL "http://9.2.216.104:8080/ogsa/services/linux/SystemStatisticsFactoryService/mystat". Below the address bar are navigation buttons: Back, Forward, New Window, Close, and Refresh. A "Go" button is next to the address bar.

The main content area has tabs for "Services", "WSDL", and "Service Data". The "Service Data" tab is selected, showing the configuration for "Linux SAR (System Activity and Reporting) Statistics".

The configuration form includes the following elements:

- Command: `sar -u`
- Interval:
- Result: OK
- Options: A b B c d l q r R u U v w W x X y
- Options: n DEV EDEV SOCK FULL
- Description: `sar -u` : Report CPU utilization

Below the configuration form is a "Grid Service" section with the following fields:

- Namespace:
- Name:
- Timeout:
- XPath Expression:
- XPath Namespace Mappings:

At the bottom of the Grid Service section are three buttons: "Query", "Subscribe", and "Unsubscribe".

The bottom part of the browser window shows an XML tree view for the service data:

```
ns1:serviceData ns1:availableUntil="2003-06-12T15:09:35.480Z" ns1:goodFrom="2003-06-11T15:09:35.480Z"
├── ns1:value xmlns:ns2="http://ogsa.globus.org/linux/SystemStatistics" xsi:type="ns2:CPUUtilizationT
│   ├── systemName xmlns=""
│   ├── interval xmlns=""
│   ├── sampleTime xmlns=""
│   ├── cpu xmlns=""
│   └── userTime xmlns=""
```

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

The screenshot displays a web browser window with the address bar showing `http://127.0.0.1:8080/ogsa/services/base/registry/ContainerRegistryService`. The browser's navigation bar includes buttons for Back, Forward, New Window, Close, and Refresh. The main content area is titled "OGSA Service Bro" and contains a search bar with the URL `http://9.2.216.104:8080/ogsa/services/linux/SystemServiceFactoryService/sendmail`. Below the search bar are tabs for "Services", "WSDL", and "Service Data". The "Service Data" tab is active, showing an "Association" section with a "resourceType" dropdown set to "LinuxProcess" and an "enumerateAssociatedIDs" button. The results list "pid2334" and "pid2325". Below this is a "status" section with "status" set to "OK" and "started" set to "true", along with "getStatus", "stopService", and "startService" buttons. The "Grid Service" section includes fields for "Namespace", "Name" (set to "ServiceState"), and "Timeout", along with "XPath Expression" and "XPath Namespace Mappings" fields, and "Query", "Subscribe", and "Unsubscribe" buttons. The bottom section displays an XML tree view with the following structure:

```
ns1:serviceData ns1:availableUntil="2003-06-12T17:24:40.966Z" ns1:goodFrom="2003-06-11T18:59:03.986Z" xmlns:ns2="http://ogsa.globus.org/linux"
├── ns1:value timestamp="2003-06-11T18:59:03.986Z" xmlns:ns2="http://ogsa.globus.org/linux"
│   ├── pid xmlns=""
│   ├── pid xmlns=""
│   └── started xmlns=""
```

Summary

- **Grid Enabled Manageability Services**
 - Persistent Heart Beat Service
 - Persistent Linux OS Service
 - Linux Processes Factory and Services
 - Disk Partition Factory and Services
 - System Statistics Factory and Services
 - Manageability Service Factory and Services for System Services
- **<http://alphaworks.ibm.com/tech/gems>** includes above 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