OSGi Service Platform in Integrated Management Environments

Telefonica I+D, DIT-UPM, Telvent
Today Management Environments

• Network Management.
  – Monitors the health and activity of the servers running back-end applications.
  – Most common protocol: SNMP (Simple Network Management Protocol)
  – Many hardware vendors provide MIB definitions and SNMP agents
  – But ... SNMP does not easily provide monitoring and management for an application

• Platform Management
  – Manage the platform itself together with the application independent functionalities (deployment, undeployment, etc)
  – Companies implementing platforms usually develop their own management tools when no standards are defined

• Application Management
  – Developers usually focus first on the service and then on management
  – When management is considered, companies develop their own tools directly for their applications
  – Developers use to write specific code in order to expose applications configurations, health, events and even statistics
Future Management Environments

• Fully Integrated Management Environment
  – Manage all network hardware, platforms and applications on top together with the same tools

• Why is it important to reach?
  – Combining health information about networks, hardware, platforms and applications will help to solve and detect problems

• How can it be reached?
  – Defining management standards and middleware to instrument both platforms and applications on top
    • Defining a generic application management framework will help to encapsulate all application manageable resources in a uniform way and expose them in a distributed environment
    • More uniform and robust application management solution
  – Integrating network protocols standards and application/platform management standards
    • Providing mechanisms for easily mapping existing management protocols such as SNMP into platform/application management framework structures
Platform & Application Management Requirements

- Platform/Application Life-cycle (State) Management
- Platform/Application Remote and Dynamic Configuration
- Platform/Application performance and statistics
- Platform/Application “health” diagnosis
- Monitoring and Debugging facilities (Logs)
- Collecting Platform/Application statistics
- Dynamic discovery of management objects and their capabilities
- Remote management:
  - Get manageable components information (attributes, operations, notifications)
  - Execute actions on manageable components
- Easy to use
- Leveraging/Integrating with existing technologies
- Flexible
  - Support several application integration
  - Support many communication protocols
OSGi Management Definitions

• Management Agent definition: “A set of one or more bundles that run on the Service Platform and communicate with a Remote Manager to provide Management of the Service Platform”

• Remote Management may include
  – Bundle life-cycle management
  – Configuration management (Configuration Admin Service)
  – Performance management (currently no specifications)
  – Fault management (Log management Service)
  – Security management (User Admin, Permission Admin)
  – Accounting management
Java Management eXtensions (JMX)

- JMX is an API for managing java applications
- Allows to expose your application components, attributes, configuration and events to management tools in a process called instrumentation
- A JMX compliant management console will see all managed resources using the same interfaces and meta data through a JMX agent
- You decide which attributes and methods you want to expose for management
- Supply a notification model, where a broadcaster service informs registers listeners with important events
- JMX is a worldwide accepted technology
- Integrates with existing network management protocols such as SNMP
- J2EE management architecture is based on Java Management eXtensions
OSGi-JMX Friends?

- OSGi is a Java based software platform
- JMX is nowadays the Java-based “de facto” management standard
- Java Management eXtensions is the basics for J2EE Management standard
- JMX Integrates with existing network management protocols such as SNMP
- JMX is a worldwide accepted technology
- Human resources and tools are ready to operate now
- JMX will be included in J2SE 1.5
- Rapid and integrated instrumentation of both applications and platforms
• However JMX and OSGi overlaps
  – JMX defines its own registry: MBeanServer
  – JMX defines its own component model: MBeans
  – JMX defines its own notification mechanism
  – JMX defines its own agent services:
    • Deployment mechanisms: M-let Agent
• Not suitable when using J2ME/CDC under OSGi
• It is not enough to support management of application deployment and application dynamic software (bundle) reconfiguration
• Approach A: Use JMX as an umbrella to manage OSGi enabled platforms
  – JMX and OSGi can be seen as complementary
  – With JMX integrated environment will be easily reached
  – JMX offer most of the functionalities that OSGi lacks such as performance management and application management instrumentation
  – But JMX and OSGi overlaps (component model, deployment model, notification model)
  – However overlapping between JMX and OSGi can be solved
  – Not suitable when using OSGi over J2ME/CDC (Mobile environments) instead J2SE (Home Gateways environments)

• Approach B: Defining a new OSGi Management Extensions for platform and applications instrumentation
OSGi Application Management: Conclusion

• Conclusion: Managing OSGi using JMX is suitable, easy to use and easy to integrate with OSGi in order to Manage Home Environments (Home Gateways)
Managing OSGi with JMX: JMX Concepts

Manager Level
- SNMP Manager
- Web Browser
- JMX Manager

Connector Level
- SNMP Adaptor
- HTTP Adaptor
- RMI Adaptor
- SOAP Adaptor

Agent Level
- MBean Server

Instrumentation Level
- MBeans
- MBeans
- MBeans

Application

© copyright 2004 by OSGi Alliance. All rights reserved.
OSGi to JMX Mappings

• Dynamic discovering and mapping of OSGi Manageable Entities: When an OSGi entity detected, an MBean or several MBBeans will be dynamically created in order to manage it through JMX
  • OSGi Framework entities: Framework, Bundles, Packages and services
  • OSGi R3 Specification Services
    – Start Level, Permission Admin, Log Service, Configuration Admin, User Admin, Package Admin, Etc.

• User Manageable Services can
  – either implements their own beans
  – or their application resources can be automatically mapped into MBBeans
  • Creating OSGi oriented application management infrastructure for OSGi service instrumentation
  • Manageable user services must be instrumented according to OSGi application management infrastructure already defined
OSGi Management with JMX: Details
Future Work

• OSGi to JMX mappings are already being defined
  – Telefonica I+D together with Telvent and DIT-UPM are still defining OSGi-JMX mappings
  – Remote JMX Management Console: Using any JMX compliant open source console (MC4J) or develop our own
  – Any contributions will be very welcome

• Managing application provisioning and application reconfiguration
  – Deploy/undeploy/start/stop applications.
  – Receive alarms and notifications
  – Query applications attributes (state, bundles, etc)

• OSGi to SNMP support
  – SNMP support will be reached through JMX infrastructure
  – A JMX-SNMP bridge will be used to map OSGi entities into SNMP
  – Integration between network management and OSGi application management will be now possible !!
  – Both network and applications remote management using an SNMP console, such as OpenNMS Console:
References

• Managing OSGi using JMX is part of ITEA Osmose Project, WP3
  – http://itea-osmose.org
  – Open source implementation
  – Authors: DIT-UPM, Telefonica I+D, Telvent

• Others:
  – OSGi Service Platform Release 3 (http://www.osgi.org)
  – OSCAR (http://oscar-osgi.sourceforge.net)
  – MX4J (http://mx4j.sourceforge.net)
  – MC4J (http://mc4j.sourceforge.net)
  – OpenNMS (http://www.opennms.org)
OSGi Related Projects

TEAHA
The European Application Home Alliance

OMS

OSO

UMS

UBISE

ePerSpace

EnComPAs

families

Jules Verne

© copyright 2004 by OSGi Alliance. All rights reserved.