FraSCAti with OSGi

Philippe Merle & Christophe Munilla
INRIA ADAM

OSGi Users' Group France, IFSTTAR, Villeneuve d'Ascq, France, 19th May 2011
Outline

- Four challenges for SOA
- SCA overview
- Combining SCA with OSGi
- FraSCAti overview
- FraSCAti with OSGi
- Related work
Four challenges for SOA

• Structuring reusable flexible architectures
  • What is behind the scene?
  • Reuse the wheel when possible…
  • …Or tune it if not!

• Interoperability
  • Orchestrate heterogeneous services
    – The weather of the city where a Twitter user lives?
    – Access the REST-based Twitter user account then a SOAP-based weather service

• Integration
  • Compose heterogeneous piece of software to build a new service
    – e.g., compose a BPEL process, an OSGi bundle and a Xquery script

• Dynamically reconfigurable runtime architectures
  • research challenge #1 in Service foundations [Papazoglou 07]
Service Component Architecture (SCA)
An OASIS’s standard programming model for SOA

- Technology-agnostic, adaptability, variability
  - Interface languages (Java, WSDL, OMG IDL, etc.)
  - Implementation languages (Java, Spring, OSGi, BPEL, C/C++, etc.)
  - Binding protocols (WS, REST, JSON-RPC, Java RMI, CORBA, etc.)
  - Non functional aspects, aka SCA intents and policies
  - Packaging formats and deployment targets (JAR, JBI, WAR, OSGi, etc.)
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
    name="bank.account">

    <service name="Account" promote="AccountFacade">
        <interface.java interface="services.account.Account"/>
        <binding.ws port="http://example.org/Account#
                wsdl.endpoint(Account/AccountSOAP)"/>
    </service>

    <component name="AccountFacade">
        <implementation.java class="services.account.AccountFacadeImpl"/>
        <reference name="StockQuote"/>
        <reference name="AccountData"
            target="AccountData/Data"/>
        <property name="currency">EURO</property>
    </component>

    <component name="AccountData">
        <implementation.bpel process="QName"/>
        <service name="Data">
            <interface.java interface="services.account.Data"/>
        </service>
    </component>

    <reference name="StockQuote" promote="AccountFacade/StockQuote">
        <interface.java interface="services.stockquote.StockQuote"/>
        <binding.ws port="http://example.org/StockQuote#
                wsdl.endpoint(StockQuote/StockQuoteSOAP)"/>
    </reference>

</composite>
Modelling with Eclipse SCA Tools
Combining SCA with OSGi

• Interoperability and integration between SCA and OSGi

  • http://www.osoa.org/download/attachments/250/Power_Comination_SCA_Spring_OSGi.pdf

• Already implementations of these ideas
  • Apache Tuscany, Fabric3, Newton

• Not still part of the SCA standard 😞
Combining SCA with OSGi

- OSGI to SCA
  - SCA service exported into the OSGi registry
- SCA to OSGi
  - SCA reference imported from the OSGi registry
- OSGI in SCA
  - SCA component implemented by OSGi bundles
- OSGi for SCA
  - Dynamic "à la carte" SCA applications
- SCA for OSGi
  - Assembly model
  - Multiple implementation languages
  - Multiple network binding protocols
  - Non functional policies and intents
OSGi to SCA

```
<component name="X">
  <service name="S">
    <binding.osgi .../>
  </service>
</component>
```
SCA to OSGi

<component name="X">
  <reference name="R">
    <binding.osgi .../>
  </reference>
</component>
OSGi in SCA

Composite bank.account

Service Account

Component AccountFacade

BPEL

Component AccountData

OSGi Bundle

Reference StockQuote

<component name="X">
  <implementation.osgi...>
</component>

© Copyright 2007 OSOA Collaboration
OSGi for SCA
OSGi for SCA
Combining SCA with OSGi
Distributed heterogeneous containers

© Copyright 2007 OSOA Collaboration
FraSCAti Overview

• A framework for SOA interoperability and integration

• A reflective SCA component model and framework
  • Runtime adaptability
  • Lightweight, efficient, predictable, scalable

• Components everywhere
  • Adaptability of all software layers

• An open source SCA implementation
  • LGPLv2 at http://frascati.ow2.org
FraSCAti, an open reflective SCA platform

Reflective FraSCAti API

Ad-hoc manual reconfiguration

Anticipated scripted reconfiguration

Middleware for integrating middleware

Interoperability

Apache CXF
SOA stack
(WS, REST)

Java RMI

JMS

JSON

RPC

OW2

PEtALS

JBI

ESB

CORBA

Integration

Java Script Engines

Spring Framework

OSGi

Apache Felix

Equinox

Easy BPEL

Engine

Easy BPEL

Engine

Easy BPEL

Engine

Easy BPEL

Engine

Easy BPEL

Engine
FraSCAti on OSGi

• Set of OSGi bundles
  • Now: few couple
  • WIP: 200+
  • Issue: Bundle granularity?

• Gateway agnostic
  • Now: Equinox, Felix, Knopflerfish
  • WIP: Concierge, …
  • Issue: Manifest portability?

• Combining SCA with OSGi
  • Now: <implementation.osgi>
  • WIP: <binding.osgi>

• Dynamic « à la carte » FraSCAti
  • WIP: Install, start, update, stop, uninstall FraSCAti plugins
Related work

• Apache Tuscany, Fabric3, Newton
  😊 More or less all presented “Combining SCA with OSGi” features

• OSGi Declarative Services
  😞 No composites

• OSGi Blueprint
  😊 Composites

• Spring Dynamic Modules
  😊 or heavy 😞?

• Apache Felix iPOJO
  😊 Very very well 😊😊 но
  😞 Will BluePrint kill it?

😞 All OSGi component frameworks
  😞 Only Java implementation
  😊 What about various remote bindings

😞 All: No fine grain reflective & reconfiguration capabilities
FraSCAti everywhere, what else?

Visit [http://frascati.ow2.org](http://frascati.ow2.org)

Contact
- Philippe Merle: [Philippe.Merle@inria.fr](mailto:Philippe.Merle@inria.fr)
- Lionel Seinturier: [Lionel.Seinturier@univ-lille1.fr](mailto:Lionel.Seinturier@univ-lille1.fr)