SOA Tools Project Charter Presentation

Carl Trieloff – IONA Technologies

SOA Tools Project Charter Overview

- SOA Tools Project (STP) provides an extensible, tooling platform for SOA applications and networks
- Key capabilities of STP include the design, configuration, assembly, deployment, monitoring, and management of SOA services
- Key specifications in STP are SCA and WSDL
- Philosophy of STP is the re-use of technology from other Eclipse projects whenever possible, such as WTP,DTP,SDO

Key Principles

Contract Independence:

- STP service contracts are independent from the implementation of the service and physical attributes

Standards

- STP contracts will comply with applicable industry standards,
- STP will support SCA and its associated implementation and binding types, also SCA will be the underlying data model for STP
- WSDL will be used to limit the scope to create the service/contract tools

Extensibility

- STP encourages third-party plug-ins and will ship with exemplary plugins / frameworks to be functional OOTB.
- STP supports an architecture, which through extensions allows anybody:
  - to provide tools for the editing, construction or configuration of services for deployment in the SOA System (SOAS).
- Any subproject of STP extending the SOAS is
  - 1) required to integrate with the meta-model and framework of the SOAS layer
  - 2) encouraged to be integrated into the service creation subproject.

Who supports STP

Charter reviewed/ supported by:

- BEA
- Compuware
- IBM
- IONA Technologies
- ObjectWeb
- Scapa
- Sybase

Additional committers from:

- EBM WebSourcing
- LogicBlaze
- Intallio
- Infravio
- SympathySoft
- And others

Participation

- The STP Core Framework
  - IONA, Sybase, ObjectWeb, IBM, BEA
  - Code contribution from Sybase & pending confirmation of contribution from another
- Service Assembly
  - IONA, Sybase, IBM
  - Code contribution from IONA
- STP support for JBI
  - LogicBlaze, EBM WebSourcing, ObjectWeb
- BPEL/ BPNM artefacts in services
  - Intallio, Scapa
  - Code Contribution from TPTP/ Intallio
- Initial committer count is expected to be between 35-40

Initial Committers

David Bosschaert, IONA
Oisin Hurley, IONA
Gregor Heine, IONA
Jack Lynch, IONA
Carl Trieloff IONA
Adrian Skehill, IONA
Eric Mitchell, IONA
Alex Chen, IONA
Howard Gao, IONA
Ciaran O'Cleirigh, IONA
David Beaurpere, IONA
Erica Mitchell, IONA
Fiona Kennedy, IONA
Freeman Fang, IONA
Joan Rohn, IONA
Johnson Ma, IONA
Mairead Melia, IONA
Alain Boulze, ObjectWeb
Christophe Ney, ObjectWeb

Naci Dai,Eteration
Fabrice Dewasme, Openwide
Rafael Marins, Fossil E-Commerce
Rob Cernich, Sybase
Karl Reti, Sybase
David Brandow, Sybase
James Strachan, LogicBlaze
Bruce Snyder, LogicBlaze
Hiram Chirino, LogicBlaze
Stefan Daume, Scapa
Antony Miguel, Scapa
Adrien Louis, EBM WebSourcing
Gaël Blondelle, EBM WebSourcing,
Daniel Berg, IBM
Ken Tam, BEA
Olivier Modic, Intalio
Ross Mason Symphonysoft
Mukund Balasubramanian, Infravio

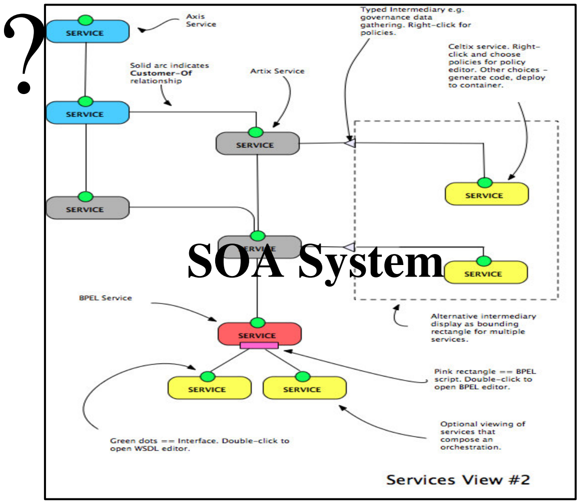
Working group topics for STP

- Core Framework [STP CF]
- Runtime Integration [STP RI]
  - Deploy-time packaging
  - Deployment
  - Management
- Service assembly [STP SA]
  - artefact creation (and editing)
  - wrt SCA subsystem
  - wrt SCA module
  - project structuring
- Infrastructure Layer [STP IL]
  - Validation
  - Refactoring
  - Reference/dependency management

- Service Discovery [STP SD]
- Testing & Debug [STP TD]
  - Management
- Extensibility Validation, how do we validate the STP frameworks, examples include
  - - POJO (All)
  - - C++ (IONA)
  - - PHP
  - - JBI (LogicBlaze, EBMWebSoucing)
  - - J2EE (IBM, BEA, ObjectWeb)
  - - BPEL (IBM, Scapa, ..)

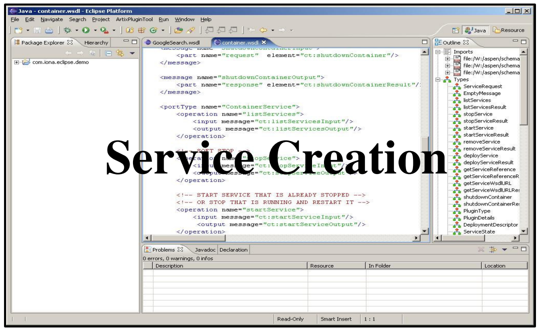
STP TLP – What does this look like?

Assembling the SOA System


SERVICE
Axis Service
SERVICE
Solid arc indicates **Customer–Of** relationship
Artix Service
SERVICE
Typed Intermediary e.g. governance data gathering. Right–click for policies.
Celtix service. Right–click and choose policies for policy editor. Other choices – generate code, deploy to container.
SERVICE
SERVICE
SERVICE
SERVICE
BPEL Service
SERVICE
Alternative intermediary display as bounding rectangle for multiple services.
Pink rectangle == BPEL script. Double–click to open BPEL editor.
SERVICE
SERVICE
Green dots == Interface. Double–click to open WSDL editor.
Optional viewing of services that compose an orchestration.
Services View #2

SOA System

Editing & creating Services



```
        <part name="request"  element="ct:shutdownContainer"/>
    </message>

    <message name="shutdownContainerOutput">
        <part name="response" element="ct:shutdownContainerResult"/
    </message>

    <portType name="ContainerService">
        <operation name="listServices">
            <input message="ct:listServicesInput"/>
            <output message="ct:listServicesOutput"/>
        </operation>

        <!-- SOFT STOP -->
        <operation name="stopService">
            <input message="ct:stopServiceInput"/>
            <output message="ct:stopServiceOutput"/>
        </operation>

        <!-- START SERVICE THAT IS ALREADY STOPPED -->
        <!-- OR STOP THAT IS RUNNING AND RESTART IT -->
        <operation name="startService">
            <input message="ct:startServiceInput"/>
             <output message="ct:startServiceOutput"/>
        </operation>
```

0 errors, 0 warnings, 0 infos

Service Creation

Policies editors

Deployment framework

System validation framework

Dependency management

B2J

BPNM

Additional Service Ext.

Frameworks

Extensibility for plug-in for specific editors

Subprojects

SOA System (STP Core Framework)

- The common core subproject defines an in-memory model that all other STP subprojects must support.
    - model will be based on the SCA assembly model schema
    - multiple extensibility requirements
    - support for "1st class" extension of SCA model, e.g. new bindings, impl type
    - support for annotations to assembly model for subprojects that want to implement runtime specific deployment

Service Creation (Service Assembly)

- Editors/links to edit/view service contracts, and implementations

B2J

- BEPL2.0 to Java code generation for consumption by Service creation plug-ins

BPNM

- Editors to edit/view BEPL2.0 artefacts of service in the SOA System

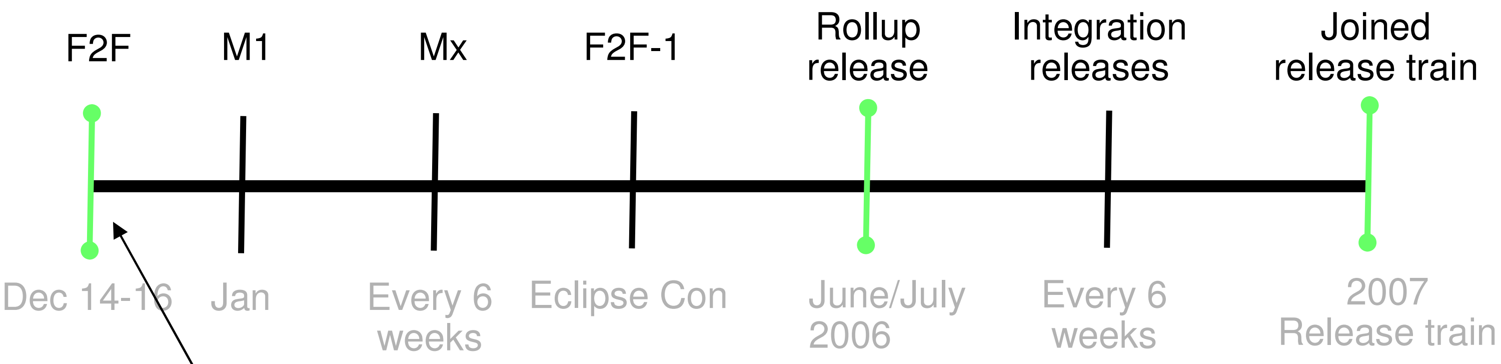
Timeline

STP Build out details

- STP will create a continuous build system, and then work on a 6 week milestone program
- STP will set up coverage stats and check-ins will be required to have 70% test coverage, for all milestone releases

- **Release train**
  - STP will not sign up for the 2006 release train, but will work to release a stable version of the frameworks and extensions within one month of the release train. This release will target:
    - STP Core Framework
    - Basic service creation (Service Assembly)
    - The Infrastructure Layer
    - Runtime Integration
    - Moved TPTP contribution to B2J code
  - STP will sign up for the 2007 release train.

Milestones



F2F
Dec 14-16

M1
Jan

Mx
Every 6 weeks

F2F-1
Eclipse Con

Rollup release
June/July 2006

Integration releases
Every 6 weeks

Joined release train
2007 Release train

M1

- Infrastructure and build systems up an running

Mx

- Incremental build working on STP frameworks and infrastructure
  - STP Core Framework
  - Basic service creation (Service Assembly)
  - The Infrastructure Layer
  - Runtime Integration
  - Moved TPTP contribution to B2J code

▪F2F-1

- Face to face at Eclipse Con

Roll up release

- STP will not join the release train but work to release all the key frameworks, in a stable tested release on top of the 2006 train.

Integration releases

- STP will continue with the 6 week cycle, and then join the release train for 2007

▪2007 release train

- STP will release a “full” version with the 2007 release train