Swordfish 0.8 Release Review

Planned Review Date: 03-18-2009

Communication Channel: eclipse.swordfish

Oliver Wolf (Project lead)

Introduction

- Swordfish provides an extensible runtime framework aimed at creating service-oriented applications
- Swordfish is internally based on Apache ServiceMix 4 as the core messaging engine
- Swordfish hooks into ServiceMix and adds functionality that is required for enterprise environments, such as service registry integration, remote configuration and monitoring

Features

- General interceptor framework that hooks into the underlying messaging engine (Apache ServiceMix NMR)
  - Message processing controlled based on meta-data carried inside or external to the message, e.g. policies
- APIs and exemplary plug-ins based on the general framework for specific areas that are significant for enterprise usage:
  - Dynamic Service Resolution: Resolve logical service endpoints into physically addressable endpoints by querying a service registry at runtime
  - Monitoring: Generate monitoring events that allow for detailed tracking of how messages are processed and that can be stored for later analysis or reporting or fed into a CEP (complex event processsing) engine (not part of Swordfish)
  - Remote Configuration: Configure framework via a local Configuration Agent that can retrieve configurations from a remote server and uses the OSGi Configuration Admin service to provide them to the framework

Features – not accomplished yet

- Some features that were originally planned for this release have not been accomplished for various reasons
  - Basic SCA support (due to time and resource constraints)
  - Integration of the Apache ODE BPEL engine (due to technical and IP issues)

Non-Code Aspects

- Sample project demonstrates message flow through the framework
- JUnit-based integration tests
- Javadoc for the framework APIs
- Basic Getting Started document available
- Documentation in Wiki is at an early stage but evolving

APIs

- All APIs are provisional and are expected to evolve based on community feedback, stabilization is planned for 1.0 release
- Extensibility and customizability is one of the key aspects of Swordfish
  - General interceptor API (creation of custom interceptors, custom processing planners etc.)
  - Service Resolver API (integration of custom service registries, custom service description document types etc.)
  - Configuration API (integration of custom configuration backends)
  - Monitoring API (integration of custom event types, event sources and event receivers)

Architectural Issues

- Swordfish makes extensive use of OSGi services in order to reduce coupling of internal components and allow for extensibility and customizability
- Plug-ins are registered using the OSGi Whiteboard pattern
  - Plug-in developers are free in their technology choice:
    OSGi DS, Spring DM, SAT, manual service registration
- All internal OSGi services are registered with a low service rank and can be replaced by custom implementations
- Swordfish core is based on Spring DM as the dependency injection framework
  - robust tracking of service registrations/de-registrations → fully dynamic behaviour, no re-starts required
  - components easily replaced by Mocks for testing purposes

Architectural Issues (cont'd)

- A potential functional overlap with the Eclipse Communications Framework (ECF) has been identified and will be resolved until the 1.0 release

Tool Usability

- The Swordfish project itself does not aim at providing any tools
- We are working closely with various sub-projects and components of STP in order to make service creation and deployment into Swordfish smooth and seamless
- For the next Swordfish release (0.9), we plan to have a first round-trip integration with STP ready

End-of-Life

- Since this is the first Swordfish release, this does not apply.

Bugzilla

Severity \ Status	NEW	ASSIGNED	RESOLVED	Total
blocker	3	.	3	6
normal	17	4	16	37
Total	20	4	19	**43**

Standards

- Parts of Swordfish's API make use of concepts from JBI (JSR 208) instead of re-inventing the wheel
- All standards relevant in the SOA space are supported through third-party components, e.g. WSDL, SOAP, WS-Security, WS-Addressing,...
- JAX-WS can be used to implement services
- Support for BPEL as a language for process orchestration is planned for the 0.9 release (via the integration of Apache ODE)

UI Usability

- Not applicable, since Swordfish currently does not include any components that expose a UI.

Schedule

- 0.8M1: planned for May 2008, delivered in time
- 0.8M2: planned for July 2008, delivered in time
- 0.8M3: planned for October 2008, slipped by 4 weeks
- Release 0.8 (formerly called 1.0) was originally planned for December 2008, but due to IP compliance issues with the Spring framework and Spring DM, the release date slipped
- Currently, the team is fully focused on the Galileo release in June 2009.

Communities

- Swordfish currently has 5 committers and 7 contributors
  - 4 committers from SOPERA, 1 from Progress Software (former IONA)
- Open and public development process
  - following the Scrum methodology
  - planning and progress tracking done in the open in a public group chat (currently on Skype, moving to IRC)
  - daily transcripts of chat posted to swordfish-dev mailing list
- Actively Evangelizing
  - Talks on EclipseCon 2008, Eclipse Summit Europe 2008 and EclipseCon 2009
  - Talks on various other conferences, e.g. JAX, Java User Group Stuttgart etc.
- Close interaction with the Apache Community
  - working with the ServiceMix 4 and ODE projects to resolve integration issues

IP Log

- All applicable IP policies and procedures as defined by the Eclipse Foundation have been followed.
- For all third-party libraries in use, the corresponding CQs have been approved.
- All source code was either 100% written by one of the committers or by a contributor who works for the same member organization as the committer.
- Swordfish's IP log can be found at http://www.eclipse.org/projects/ip_log.php?projectid=rt.swordfish
- A frozen copy of the reviewed-and-approved-by-Eclipse-legal IP log has been supplied as part of the Release Review documentation

IP Issues

- As to our knowledge, there are currently no unresolved IP issues related to this release.