Swordfish 0.9 (Galileo) Release Review

Planned Review Date: 06-10-2009

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
- Swordfish includes basic tool support and additional components such as a Service Registry and a Process Engine

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 – cont'd

- Integrated process engine capable of executing BPEL processes (Apache ODE)
- Basic tools supporting the most important use cases
- Service Registry to dynamically resolve logical service names into service endpoint addresses

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)

Non-Code Aspects

- Sample project demonstrates message flow through the framework
- Unit test coverage has improved since last release
- JUnit-based integration tests have been added
- Tool components are well documented (Eclipse Help, Cheat sheets)
- Documentation for runtime parts in Eclipse Wiki has significantly improved since last release, but is still evolving
- Javadoc for the framework APIs
- Basic Getting Started document available

APIs

- All APIs are still provisional and are expected to further 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)
  - based on open standards (such as JBI)

Architectual 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

Architectual 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
- The Apache ODE process engine is also used by the SMILA project (not part of Galileo). We're actively working with the SMILA team to avoid any duplicated effort for bundling, adaption to EclipseLink etc. on either side.

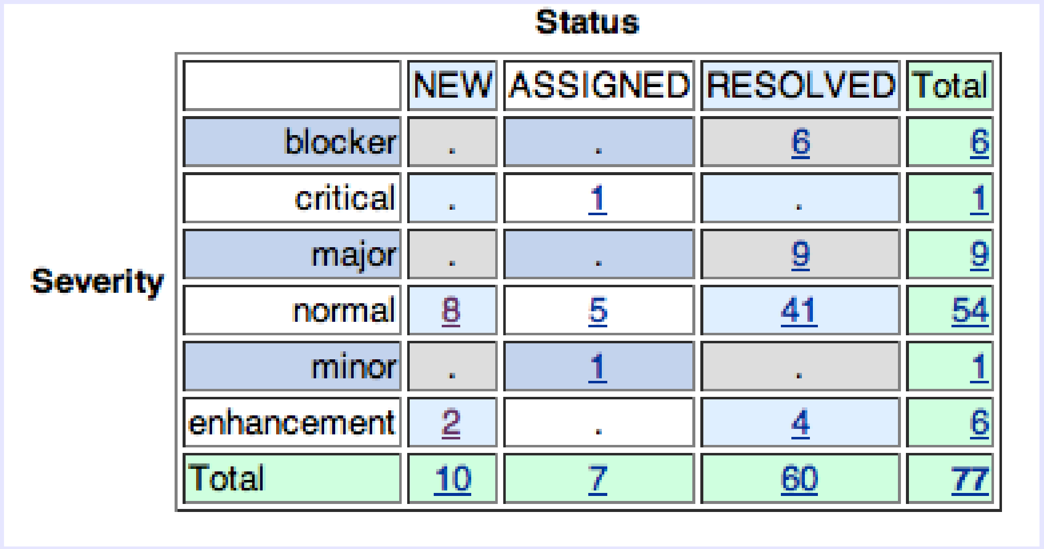
Tool Usability

- Swordfish provides basic tooling to support the most relevant service development use cases:
  - Implement services based on a WSDL document (WSDL-first approach)
  - Implement services based on a Java interface (Code-first approach)
  - Publish WSDL documents into the service registry
- For process development and deployment, Swordfish relies on the Eclipse BPEL Editor which we have adapted to work with Galileo
- We are working closely with various sub-projects and components of STP in order to make service creation and deployment into Swordfish even more smooth and seamless

End-of-Life

- No features from the previous release have been end-of-life'd in Release 0.9.

Bugzilla



Status

Severity	NEW	ASSIGNED	RESOLVED	Total
blocker	.	.	6	6
critical	.	1	.	1
major	.	.	9	9
normal	8	5	41	54
minor	.	1	.	1
enhancement	2	.	4	6
Total	10	7	60	77

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 and is specifically supported by the Swordfish tools
- BPEL is used as the process orchestration language (via the integration of Apache ODE)

UI Usability

- The Swordfish tool components expose only a very limited UI which adheres to the Eclipse User Interface Guidelines. Most UI elements are actually reused from the Workbench.

Schedule

- Release 0.8: delivered 03/04/2009 (was originally planned for December 2008, but due to IP compliance issues with the Spring framework and Spring DM, the release date slipped)
- 0.9 M1: planned for 07/04/2009, delivered in time
- 0.9 M2: planned for 29/04/2009, delivered in time
- 0.9 M3: planned for 19/05/2009, delivered in time
- 0.9 RC1: planned for 27/05/2009, delivered in time
- 0.9 RC2,3: planned for 03/06/2009, 10/06/2009
- 0.9 GA: planned for 17/06/2009

Communities

- Swordfish currently has 6 committers and 6 contributors
  - 5 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)
  - Product and sprint backlogs are published to the Wiki to solicit feedback from the community
- Actively Evangelizing
  - Talks on EclipseCon 2008, Eclipse Summit Europe 2008, EclipseCon 2009 andvarious 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.