Riena Release Review 1.0

Project Lead: Christian Campo
(Christian.Campo@compeople.de)

Riena Mailing List: riena-dev@eclipse.org

IP Log: http://www.eclipse.org/projects/ip_log.php?projectid=rt.riena

January 21, 2009

Agenda

- Features
- Non-Code Aspects
- APIs
- Architectural Issues
- Tool Usability
- End-of-Life
- Bugzilla
- Standards
- UI Usability
- Schedule
- Communities
- IP Issues
- Future plan

Riena - Mission

The Riena platform is a foundation for building multi-tier enterprise client/server applications.

As such Riena broadens the usage of the service oriented architecture of OSGi/ Equinox by providing access to local and remote services in a transparent way.

The Riena user interface components supports an enduser focused navigation metaphor which can be used as an alternative to the default workbench layout.

Riena builds new or leverage existing components that are need to implement client/ server applications based on Eclipse technology.

Features

- Distributed Enterprise Applications
  - Remote OSGi Service based on Webservice model
- Enhanced UI Model based on RCP
  - Based on RCP but with a different look and navigation model (enduser focused)
  - separation between controller/view using Ridget API
  - Validator, Converters, Markers, UI Filters
- Riena Core
  - Injector for services and extensions
- Security
  - using equinox security
  - supports authentication and authorization (client and server)
- Object Transaction
  - follows object net modifications in a distributed environment
- Monitoring
  - transfer logs, statistics etc from client to server
- New and Noteworthy → http://wiki.eclipse.org/Riena_New_And_Noteworthy

Features (not accomplished yet)

- Not started due to missing time and resources
  - Reporting
  - Software Update / Provisioning
  - Persistence
- all intended to leveraging existing eclipse projects

Non-Code Aspects

- Riena Wiki pages contains already many howtos for a number of aspects of Riena
- 9 sample/demo projects show various aspects of Riena
- snippet project (and wiki page) that show how to use Ridgets
- Many JUnit TestCases (1366 on last count)
- Javadoc
- Wizard that can generate Riena UI samples

APIs

- All APIs are provisional since we still expect to learn from our community. We plan to solidify the APIs for the 1.1. Release.
- Many Testcases (over 1300) to validate that APIs are working
- Many extension and services where users can replace Riena functionality with their custom implementation.
- Injector API (inject services and extensions)
- Remote Service API (access remote services)
- Ridget API (separate controller/view in the UI, simplifies use of widgets)
- Navigation API (for the navigation model)
- ObjectTransaction API
- Look & Feel API (to extend the UI with a custom themes and looks)
- UI Filter API (to manage restrictions in the UI)

Architectural Issues

- We use OSGi Services wherever that is possible.
  - easily consumable by others
  - replaceable by custom services
- We use Extensions to make Riena extendable by others
- Riena can be consumed in pieces
  - You can only use Remote Services or only use Ridgets or only use Object Transaction.
- OSGi Services of Riena are registered with a low Service Rank so that they can easily be replaced by a custom solution (OSGi Services with a default or higher rank).

Architectural Issues (2)

- Remote Services are implemented in Riena based on the Hessian Protocol. The goals and motivation for our current implementation are documented here →http://wiki.eclipse.org/Riena_RemoteServices
- We had numerous face-to-face, email, mailing list etc. discussions with Scott Lewis from ECF how and if Riena Remote Services overlaps with ECF Remote Services.
- Because of the different target audience we (Riena) think that there is some overlap between the two (i.e. based on OSGi Services) but also a number of differences in the approaches (transparent vs explict call, connection-oriented vs connection-less etc.)
- Very good feedback from the community using Riena Remote Services.
- We intend to work together with ECF on a RFC 119 compliant API. So that users can program against one API and either use ECF or Riena inside.

Architectural Issues (3)

- Dependency Injection used in all placed for services and extensions using the Injector APIs
  - making components themselves independant of the Injector API
  - make them easily testable using Mocks
  - automatically tracked coming and going services and extensions

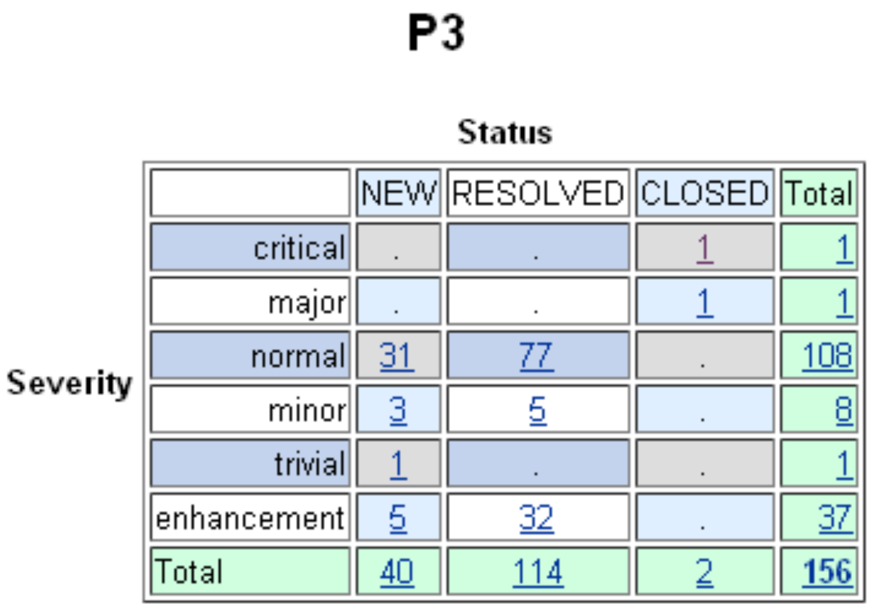
Tools Usability

- Riena currently has no tools
- We have a Wizard which can be used to generate very simple sample client applications.

End of Life

- Does not apply since that is the first release.

Bugzilla



P3

Status

Severity	NEW	RESOLVED	CLOSED	Total
critical	.	.	1	1
major	.	.	1	1
normal	31	77	.	108
minor	3	5	.	8
trivial	1	.	.	1
enhancement	5	32	.	37
Total	40	114	2	156

Standards

- Attempt to base as much as possible using OSGi standards
- Remote Services are based on the concept of Webservices

UI Usability

- Built on top of Eclipse RCP
- Use proven technologies from their like Views, Perspectives
- Adds a different navigation model on top that is more driven by the requirements of endusers.
- Ability to customize Look and Feel for SWT views for various use cases and customers.

Schedule

- Started in October 2007
  - Initial Schedule (as specified in the proposal):
    - 2008-01 Version 0.1
    - 2008-04 Version 0.4
    - 2008-07 Version 0.7
    - 2008-10 Version 1.0
- First milestone was in March and then then roughly in 2 month intervals.
- 1.0.0.M5 in November, 1.0.0.M6 in December
- Downloads as ready to run platforms (including Eclipse RCP and Equinox SDK) for Windows, Mac OSX and Linux or as Riena only download zip.
- All milestones were released in time.

Process: Community and Activities

- 10 committers currently, 2 contributors that are activly submitting patches.
- Open communication via Riena newsgroup
  - Rapid response on questions mostly by Riena committers (sometimes by Riena users)
  - Plans available on Riena wiki
  - open discussion among committers in riena-dev maillist
- Open and inclusive release planning and tracking processes
  - Bugzilla used to request and track all defects and enhancements
  - Bugzilla entries from committers and Riena users
- All contributions made directly to Eclipse CVS
  - code written by comitters
  - code from contributers enter through bugzilla patches
  - code from the initial code contribution from compeople entered through IPZilla
- Coordination/cooperation with other Eclipse projects
  - equinox.log in a new extended version from equinox incubator is used
  - dependency to Nebula existed and was later removed
  - interaction with EclipseLink and Birt has been discussed internally
  - interaction with ECF was mentioned above and discussed on maillinglist and newgroup

Process: Community

- Evangelism and outreach into the community
  - Presentations on Eclipse Summit and Eclipse Con and Jax
  - Multiple articles in german Eclipse Magazin and Java Magazin.
- Downloads per milestone
  - M3 (941), M4(699), M5(414), M6 (346), RC1(54)
- Riena is often consumed in pieces
  - people start with Remote Service
  - then they try Ridgets
  - then the whole UI with the alternative navigation model
- Ekke Gentz is very active (http://ekkes-corner.blogspot.com) who blogs and posts a lot about using Riena for a large-scale application.
- Joel Rosi-Schwartz just recently converted ORMF to be based on Riena Remote Services
- compeople based 2 Prove of Concepts on Riena in 2008 and is starting new projects in 2009 based on Riena.
- Riena is already in production in at least one case despite its early stage

IP Issues

- Compeople offered an initial code contributation which was submitted to IPZilla.
- Riena also uses some third party software that was submitted through IPZilla.
- Everything else was written by one of the committers himself or a patch was applied from a bugzilla request.

Future plan

- Riena plans to release a 1.1. version of Riena within Galileo.