



Graphiti 0.9.0 Release Review

Planned Review Date: June 2012
Communication Channel: eclipse.graphiti
Christian Brand
Matthias Gorning
Tim Kaiser
Jürgen Pasch
Michael Wenz

Introduction



Graphiti is an Eclipse-based graphics framework that enables the fast and easy creation of graphical tools, which can display and edit underlying domain models using a tool-defined graphical notation. Graphiti supports the developer in getting to a first version of an editor with very low effort by:

- Hiding platform specific technology (e.g. GEF / Draw2D on Eclipse)
- Providing rich default implementations inside the framework
- Providing a default look and feel that was designed in close co-operation with usability specialists

Features (I)



- Make Graphiti fit for being part of the release train and eventually leaving incubation
 - Graphiti is part of the Eclipse release train for Juno since M1
 - All requirements for being part of the release train were fulfilled (see separate slide)
 - Infrastructure related topics like the migration to Git were taken care of
- Round-offs and ease of use
 - A lot of round-offs for framework functionality easing the development of Graphiti-based editors have gone into this release. Especially the Diagram Editor API has undergone a major redesign. The documentation has been improved in many areas.

Features (II)



- New functionality and round-offs
 - Introduced new connection types: curved and composite connections
 - Introduced border and color decorators
 - Added wizard to create a new plugin containing a basic Graphiti editor
 - Reworked Styles concept
 - Allowed to fully remove the palette from the editor
 - Added unconditional editor refresh
 - Added state callbacks during connection handling
 - Allowed nested shapes in anchors
 - Allowed decorators for connection texts
 - Allowed scaling of images
 - Introduced a custom type for direct editing
 - Allowed undo/redo across editor save
- For the full list of new features and implementation details see [Graphiti New & Noteworthy page for 0.9.0](#)

Non-Code Aspects (I)



- User documentation is available as part of the Graphiti SDK installation from the update site
- A large part of this documentation is a tutorial that introduces users step by step to the most common features of Graphiti
- The tutorial and documentation have been updated to reflect the changes done within the framework
- A new example (Chess diagram) has been added
- Some further examples are part of our tests and can be synced from Git

Non-Code Aspects (II)



- Website at <http://www.eclipse.org/graphiti>
- Graphiti can be downloaded via p2 update sites and as a zipped version
- Available features
 - Graphiti Feature (the framework)
 - Examples Feature (includes examples, doc plugin with eclipse help integration and javadoc)
 - Export Feature (diagram export to SVG)
 - Tools Feature (tools that can be used to create Graphiti-based editors)
 - SDK Feature (framework, examples, additionally includes sources to enable debugging)
 - SDK+ Feature (additionally includes optional parts, currently the SVG exporter)

Non-Code Aspects (III)



- Graphiti uses a Buckminster- and Hudson-based build and test infrastructure
- The framework plugins are provided with dedicated test plugins checking their valid behavior
- Unit and SWTBotTests are executed as a part of the Buckminster Build on the Eclipse [Hudson](#)
- Each new build is tested at least with Eclipse 4.2 and 3.8 (Juno), other test environments are Eclipse 3.7 (Indigo)
- Graphiti is part of the Release Train since Eclipse Indigo (M4)

APIs



- All non-API code is in « internal » packages
- APIs are consolidated and are high quality including API contract in form of Java doc
- API will further evolve based on community input
- Balance between hiding GEF/Draw2d from clients to reduce complexity and opening up the API to be heavily customizable

Architectural Issues



- Graphiti's architecture is solid and basically unchanged since the basis of the framework was already productive within SAP AG for several years
- Nevertheless several reworks have been done in the 0.9.0 timeframe to improve Graphiti's architecture to better support an open communities' varied requirements
- Some aspects with regards to API rework are still open (e.g. in the rendering area) and will be targeted for the next release

Tool Usability



- Tutorial supports getting started
- Good results in tool building can be achieved early
- Short turnaround cycles support incremental development
- Ability to debug enables fast problem solving (compared to frameworks based on code generation)
- Test preferences allow user to enable additional means to identify and analyse issues besides debugging

End of Life Issues / Changed APIs



- The diagram editor API has undergone a major rework
 - Before clients were forced to use internal classes
 - Introduced better structuring along main concerns
 - For details see [Bug 336488](#)
- Background color attribute is now adhered
 - Involves an automated migration of existing diagrams
 - See [Bug 360800](#)
- Some minor API clean-ups (typo removals and renames) in
 - DefaultToolBehaviourProvider
 - Disabling of guides inside the editor
- Otherwise no deprecated or removed features

Bugzilla



- Bugzillas during 0.9.0 timeframe
 - Opened: 145
 - Closed: 127
 - Deferred: 9
- Outstanding (mainly enhancements):
 - P1: 0
 - P2: 0
 - P3: 17
 - P4: 53
 - P5: 5
- [All open Graphiti issues in Bugzilla](#)

Standards / Execution Environment



- Execution Environment JavaSE 1.5, but also runs on Java 6 and Java 7
- Supports Eclipse Platforms 4.2, 3.8 and 3.7
- We will support Eclipse 3.8 and 4.2 equally, and all the functionality will be the same
- Based on Draw2D/GEF and SWT, de-facto standards for graphics in Eclipse
- Integrated with and based on EMF, de-facto standard for modeling in Eclipse
- No native components, therefore there are no further requirements to the execution environment

Simultaneous Release Specific



- Build offset: +3
- [Project plan](#)
- [IP Log](#)
- Direct communication: Michael Wenz, Matthias Gorning
- No deviations for simultaneous release requirements
- [New & Noteworthy](#)
- Retention policy
 - Released versions will be kept forever.
 - Final milestones (announced via mail and the downloads page on the Graphiti page) for an upcoming release will be kept until the final release is available.
 - Nightly (dev) builds might be removed without prior notice. The same is valid for milestone builds that happen each day on the way towards each milestone.
 - All source code will be kept in the Eclipse repository (Git, /gitroot/gmp/org.eclipse.gmp.graphiti)

User Interface Usability



- Usability
 - UI designers were part of the initial development of the framework at SAP AG
- Eclipse User Interface Guidelines are adhered to
- Globalization is supported
 - All strings in separate resources
 - Graphiti is enabled for translation via the Babel project
- Graphiti Editors can be used via keyboard only and support displaying and editing diagrams while the underlying operation system runs in high contrast mode (or any other mode that eases handling for disabled people)
- Screen readers are currently not supported

Schedule



- Graphiti is part of the Juno release train since M1
- All milestones since then have been met
- Because of a major reduction of the team capacity, Graphiti had to adapt the original schedule for M4
- The Graphiti team achieved all the main targets of the adapted scope for this release
- 4 topics needed to be dropped during the release
 - [Bugzilla 351193](#): Provide a test infrastructure for an RCP scenario
 - [Bugzilla 352097](#): Better support for layouts: provide standard layouts
 - [Bugzilla 352115](#): Complex tooltips
 - [Bugzilla 352120](#): Diagrams in views



Communities (I)

- [Bugzilla](#)
 - Bugs and enhancements are tracked using Bugzilla
 - User feedback and enhancement request received
 - Many contributions (bug fixes and enhancements) received
- [Newsgroup](#)
 - Is used as active communication channel (~250 threads since last release 0.8.0)
 - Many detail questions, discussions and enhancement ideas start there
- Conferences
 - Talk at EclipseCon Europe 2011
 - Presentations at Eclipse Demo Camps both for Indigo and Juno – talk is agreed for Bonn, Walldorf and Vienna

Communities (II)

- Conferences (cont.)
 - Several talks at EclipseCon 2011/12 referred Graphiti
- Eclipse projects using Graphiti
 - [eTrice](#)
 - [JPA Editor](#)
 - [CDO/Dawn](#) integrates with Graphiti
 - [BPMN2 Modeler](#)
 - [GMF tooling](#) expressed interest in supporting Graphiti as alternative runtime
- Other open source projects
 - [Activity BPM Platform](#) offers an Eclipse Designer built on Graphiti
 - [The KIELER research project](#) integrates their layouting algorithms with Graphiti
 - [Spray](#) supports a generative approach on top of Graphiti

IP Log



- The Eclipse IP policies and procedures have been followed
- The Graphiti IP Log can be found at http://www.eclipse.org/projects/ip_log.php?projectid=modeling.gmp.graphiti
- A frozen version is available at <http://www.eclipse.org/graphiti/iplog/0.9.0.pdf>

IP Issues



The EMO explicitly asks during the Release Review if any Member would like to assert that this release infringes their IP rights.

If so, the EMO and the project will follow the Eclipse IP Policy in discussions with that Member.

Credits and Kudos



- Thanks to the community for the most valuable discussions and feedback in the forum and in bugzillas
- Also Kudos to all who contributed to the project, be it in form of feedback, suggestions, questions or most valuable code contributions
- Special thanks to:
 - Felix Velasco
 - Benjamin Schmeling
 - Hernan Gonzalez

For providing a great amount of high-quality contributions