



Graphiti 0.8.0 Release Review

Planned Review Date: June 2011
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
 - Graphiti joined the Eclipse release train for Indigo in M4
 - Some gaps in globalization and usages of non-APIs from other projects were closed
 - All requirements for being part of the release train were fulfilled
- Ease of use
 - Analysis of originally planned enhancement with DrawPatterns was done but solution turned out to be too complex to be implemented in the 0.8.0 timeframe. [Bugzilla 323360](#) was postponed.

Features (II)



- Functional round-offs
 - Enablement of connections on connections
 - Management of fonts
 - Support for non-EMF domain model undo/redo
 - Deletion of multiple objects leads to only one popup
 - Patterns with remove, delete, and direct editing
 - Full themes support (accessibility)
 - Direct editing: user defined selection context, improved representation and error handling
 - Saving diagram in vector format
 - Support for URIEditorInput
- For the full list of new features and implementation details see [Graphiti New & Noteworthy page for 0.8.0](#)

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
- Some further examples are part of our tests and can be synced from the CVS
- Article on Graphiti has been published in German Eclipse Magazin 1/11
- Same article in English will be available on JAXenter.com soon

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)
 - SDK Feature (framework, examples, additionally includes sources enabling debugging)
 - SDK+ Feature (additionally includes optional parts, currently the vector graphics 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 3.7 (Indigo), other test environments are Eclipse 4.1 and Eclipse 3.6 (Helios)
- Graphiti is a new participant of the Indigo Release Train

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 smaller reworks have been done in the 0.8.0 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 editor API 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)
- Usability seems to be good in general, we quote



End of Life Issues



- Fonts are now managed inside the diagram
 - Same handling as colors
 - Introduced *manageFont* method and removed direct creation methods
 - Old diagrams with specific fonts set directly at a text object are still readable but will need conversion (migration utility is available) to preserve those fonts
 - For details see [Bugzilla 323358](#)
- Some minor API clean-ups (renames, parameter adding and removal of not needed methods) in
 - Feature for resizing shapes
 - Save as image and print functionality
- Otherwise no deprecated or removed features

Bugzilla



- Bugzillas during 0.8.0 timeframe
 - Opened: 92
 - Closed: 91
 - Deferred: 2
- Outstanding (mainly enhancements):
 - P1: 0
 - P2: 0
 - P3: 26
 - P4: 14
 - P5: 4
- [All open Graphiti issues in Bugzilla](#)

Standards



- Execution Environment JavaSE 1.6
- Supports Eclipse Platforms 3.6 and 3.7
- 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

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)

Schedule



- Graphiti is part of the release train since M4
- All milestones since then have been met
- Graphiti met the original schedule and the team achieved all the main targets for this release
- Two minor topics needed to be dropped during the release
 - [Bugzilla 323352](#): Selection menu on drag&drop of an object from the palette
 - [Bugzilla 323360](#): Introduce draw patterns

Communities (I)



- [Bugzilla](#)
 - Bugs and enhancements are tracked using Bugzilla
 - User feedback and enhancement request received
 - Several contributions (bug fixes and enhancements) received
- [Newsgroup](#)
 - Is used as active communication channel (more than 100 threads since last release)
 - Many detail questions and enhancement ideas start there
- Conferences
 - Talk and poster presentation at EclipseCon Europe 2010

Communities (II)

- Conferences (cont.)
 - Some talks at EclipseSummit 2011 referred Graphiti
- Well-received presentations held at
 - Eclipse demo camps in Walldorf and Dortmund
 - Eclipse Modeling Day in Zürich
- Eclipse projects using Graphiti
 - [eTrice](#)
 - [JPA Editor](#)
 - [Dawn](#) plans to integrate with Graphiti
 - Project Proposal [BPMN2 Modeler](#)
- Other open source projects
 - [Activity BPM Platform](#) offers an Eclipse Designer built on Graphiti
 - [The KIELER research project](#) integrates their layouting algorithms with 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/0.8.0.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