GEF Release Review 3.1

Pratik Shah
GEF Developer
IBM, Raleigh

June 22, 2005

Agenda

- Features
- Non-Code Aspects
- APIs
- Architectural Issues
- End-of-Life
- Bugzilla
- Standards
- Schedule
- Process
- Community
- IP Issues
- Project Plan

New features in 3.1

- Enhanced Text Package
  - Display rich text
  - Now supports rich BiDi content
  - Now supports rendering on mirrored Controls (e.g. Arabic)
- New support for using SWT’s advanced graphics
- Support for mirroring on Windows
- New examples
  - Shapes
  - WYSIWYG text editing
  - Graphical Ecore editing (EDiagram)

New Features in 3.1 - Continued

- Other enhancements
  - ShortestPathConnectionRouter
  - Scroll-wheel support
  - System clipboard
  - Direct-editing improvements (keyboard shortcuts, better positioning)
  - Improved performance
    - DirectedGraphLayout
    - FocusTraverseManager
    - Figures
  - Multifarious marquee tool
  - Easier to customize tools via properties

Non-Code Aspects

- Non-javadoc documentation continues to improve
  - Two new eclipse.org articles contributed by the community
    - *Building A Database Schema Diagram Editor with GEF* by Phil Zoio
    - *A Shape Diagram Editor* by Bo Majewski
  - Several unofficial wiki articles
  - Gotchas!
  - Updated, comprehensive documentation page
    - http://www.eclipse.org/gef/reference/articles.html
- Localization/Externalization
  - Continue to follow 3.0 practices; same as Platform
  - Translation verification test-cases updated
- Accessibility checklist kept up-to-date
  - http://www.eclipse.org/gef/developer/accessibility.html
- FAQs could use an update
- No other GEF-maintained material identified as obsolete
  - Javadoc and ReadMe will be updated

API

- API is Eclipse Quality
- Changes
  - org.eclipse.draw2d.**text** package
    - Was marked as not intended to be sub-classed
    - Breaking API change for FlowBoxes (fragments)
  - SWTGraphics.State
    - Likely not used by clients, but a breaking change nonetheless
    - Necessary to support advanced graphics features
  - Changes were made as early as possible

Architectural Issues

- Core architecture is stable; no changes there
- Draw2d’s text package was greatly improved
  - Quite a few architectural enhancements
    - Better functionality and scaling
  - Minimal impact on performance when BiDi is not required
- GEF introduces non-critical “new” features as internal or in examples
  - Gives us a chance to address any problems, revamp architecture if necessary, gather usage data

End-of-Life

- The following items were deprecated
  - org.eclipse.gef.GEF – Unused; most of it had been deprecated in 3.0
  - FigureUtilities#getGC() – Clients could “mess up” the GC with the new advanced graphics functionality
  - Protected fields in Figure/EditPart – To address memory consumption we would like to redesign the way rare functions are used
  - Figure#fireMoved() – To allow for CoordinateListener; needed to distinguish between figure’s bounds changing and figure’s absolute location changing (which may happen without the bounds changing)

Bugzilla

- New bugs reported since 3.0.1 – 98 (1 critical)
- Bugs fixed in this release – 118 (1 critical)
- Existing P1s and P2s – 0
- Bugs outstanding – 191
- Aside: GEF team reported >200 real bugs against Platform during 3.1
    - 3 blockers, 5 criticals
    - >70 fixed in 3.1

Standards

- GEF doesn’t follow any specific standard per se; N/A
- Mimic *implicit* standards by other graphical applications
  - Examples
    - Rulers like Microsoft Word
    - Rich-text rendering behaviour copied from browsers Firefox and IE
- Follow OS and Eclipse platform standards, where applicable
  - Examples
    - Keybindings
    - PaletteCustomizerDialog looks like platform’s Preferences dialog

Schedule

- Schedule closely followed the platform’s release schedule
  - With at most a week’s leeway
- M6 slipped by about 10 days because of build issues
  - Switched to JARred plug-ins
  - Bugs in PDE and base builder

Process

- Conformed with Eclipse standards
  - Open, transparent and inclusive
  - Newsgroup remains the main medium for clients to share knowledge, ask questions, and identify problems
  - Mailing list used extensively to elicit requirements, keep GEF shareholders updated about new developments, discuss architectural enhancements and API changes, and solicit contributions
  - Bugzilla used for tracking changes and accepting contributions
- Room for improvement
  - Detailed milestone plans
  - Follow platform process more closely
  - Provide summary of changes between milestones

Community

- GEF is becoming highly popular!
  - De facto framework for graphical-editing in Eclipse
  - Most attended tutorial at EclipseCon 2005
  - One of the most active newsgroups after platform
  - GEF is often among the Top 10 Downloads at eclipse.org
- More contributions during 3.1 than ever before
  - Shapes example (bug 71532)
  - GridLayout (bug 71684)
  - Orthogonal router (bug 71498)
  - EDiagram example properties (bug 84910)
- Community's making up for the lack of non-javadoc documentation
  - Articles, wiki, RedBooks, books, examples

IP Issues

- Successful transition from CPL to EPL
  - Updated about and license files
  - All non-committer contributors re-contributed code under the new license
  - All such contributions were approved by the Eclipse Foundation’s legal staff
  - Copyrights updated in all files (contributions acknowledged)

Project Plan

- 3.2 plan hasn’t been drawn up yet
- Looking forward:
  - Editing of rich text
  - Palette Keybindings
  - Complex layouts (grid and perhaps table)
  - Smart orthogonal routing of connections
  - Enhanced snapping support (creation, bendpoints)
  - Support for platform’s operation history framework
  - Curved connections

That’s all, folks!

- Any questions?