GEF Release Review
3.2

Steven Shaw
GEF Developer
IBM, Ottawa

July 6, 2006

Agenda

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

New features in 3.2

- Enhanced layout support

    - Ability to handle disconnected graphs in the directed graph layout algorithm.

    - New animated layout capability in any figure container.

- Improved examples

    - WYSIWYG text editing fixes for caret placement

    - WYSIWYG text editing support for key board navigation (page-up, page-down, home etc.)

- Improved performance

    - Avoiding redundant thumbnail refreshes

    - Resizing palette performance

Non-Code Aspects

- New GEF Wiki created
    - Comprehensive content added
        - http://wiki.eclipse.org/index.php/Graphical_Editing_Framework
    - Updated various FAQs

- Localization/Externalization
    - Adopted ICU4J as per platform requirement
    - Translation verification test-cases updated

- Accessibility checklist kept up-to-date
    - http://www.eclipse.org/gef/developer/accessibility.html

- Increased JUnit / automated testing coverage

- No other GEF-maintained material identified as obsolete
    - Javadoc and ReadMe will be updated

Release engineering

- Satisfying Callisto requirements
  - Update Require-Bundle of plugin manifest to specify bundle-version
  - Update build naming convention - Build Id was updated to the new naming convention adopted by platform
  - Created new update sites for GEF
    - http://download.eclipse.org/tools/gef/releases/update-site/site.xml
    - http://download.eclipse.org/tools/gef/milestones/update-site/site.xml

API (Draw2d)

- API is Eclipse Quality

- New classes / interfaces (refer to JavaDoc for further info)
    - org.eclipse.draw2d.Animation
    - org.eclipse.draw2d.Animator
    - org.eclipse.draw2d.LayoutAnimator
    - org.eclipse.draw2d.NativeGraphicSource
    - org.eclipse.draw2d.RoutingAnimator
    - Org.eclipse.draw2d.RoutingListener

API (Draw2d)

- New methods (refer to JavaDoc for further info)
    - org.eclipse.draw2d.geometry.PointList#reverse
    - org.eclipse.draw2d.geometry.PointList#transpose
    - org.eclipse.draw2d.graph.DirectedGraph (new getters / setters)
    - org.eclipse.draw2d.graph.Edge (new getters / setters)
    - org.eclipse.draw2d.graph.Node (new getters / setters)
    - org.eclipse.draw2d.graph.ShortestPathRouter#getSpacing / setSpacing
    - org.eclipse.draw2d.ShortestPathConnectionRouter#getSpacing / setSpacing
    - org.eclipse.draw2d.DeferredUpdateManager#sendUpdateRequest
    - org.eclipse.draw2d.FigureUtilities#isAncestor
    - org.eclipse.draw2d.PolylineConnection#addRoutingListener
    - org.eclipse.draw2d.UpdateManager#performValidation

API (GEF)

- API is Eclipse Quality

- New classes / interfaces (refer to JavaDoc for further info)
    - org.eclipse.gef.editparts.SimpleRootEditPart
    - org.eclipse.gef.SelectionManager.

API (GEF)

- New methods (refer to JavaDoc for further info)

    - org.eclipse.gef.commands.CommandStack#notifyListeners

    - org.eclipse.gef.commands.CommandStackEvent#isPreChangeEvent / isPostChangeEvent

    - org.eclipse.gef.editpolicies.GraphicalNodeEditPolicy#getDummyConnectionRouter

    - CombinedTemplateCreationEntry: new constructors

    - DirectEditManager: new constructor

    - org.eclipse.gef.ui.palette.FlyoutPaletteComposite#createFlyoutPreferences

    - org.eclipse.gef.EditPartViewer#getSelectionManager

    - Org.eclipse.gef.SnapToGeometry#getLocation / getType

Architectural Issues

- Core architecture is stable; no changes there

- New listeners added to allow clients to hook into key architectural events (layout, etc).

- 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

- No classes / methods were deprecated in GEF 3.2

Bugzilla

- New bugs reported since 3.1.2 – 93 (7 critical)

- Bugs fixed in this release – 73 (5 critical)

- Existing P1s and P2s – 0

- Bugs outstanding – 151

- Enhancement requests outstanding – 35

Standards

- GEF doesn't follow any specific standard per se; N/A

- Mimic *implicit* standards by other graphical applications

  - Examples

    - Rulers similar to common word processor applications
    - Rich-text rendering behaviour modeled from popular browsers

- 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
    - http://www.eclipse.org/gef/project-info/summary.html

- RC1 was 3 days late because of Eclipse.org download issues

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 continues to be highly popular!

    - De facto framework for graphical-editing in Eclipse

    - Lots of interest at EclipseCon 2006

    - Newsgroup is very active

    - GEF is often among the Top 10 Downloads at eclipse.org

- Part of Callisto effort

- Community's making up for the lack of non-javadoc documentation

    - Articles, Wiki, RedBooks, books, examples

IP Issues

- Fixed remaining erroneous occurrence of CPL license in source and changed to EPL.

Project Plan

- 3.3 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
    - Enhanced image support (copy / paste / export / display of formats)

That's all, folks!

- Any questions?