G.E.F.

Past, Present & FUTURE

Alexander Nyßen
itemis AG

Past

It all began...

Image: winnond / FreeDigitalPhotos.net

... at IBM, of course (etools).
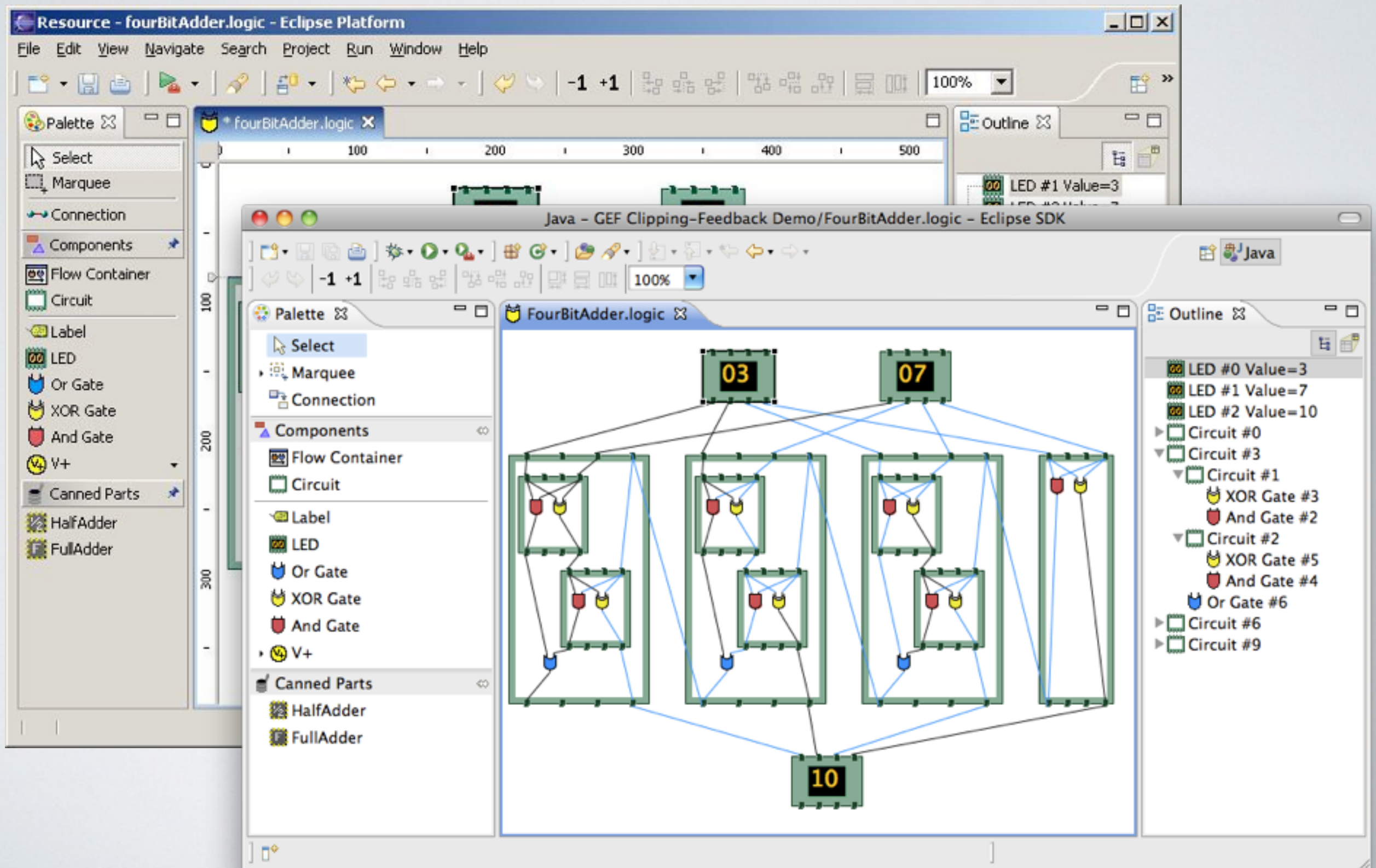
It's Birthday Time!

Draw2d & GEF (MVC)

- Initial contribution of Draw2d & GEF (MVC) by IBM in 2002.

Draw2d - 2D rendering framework; lightweight extension to SWT. May be used stand-alone or as visualization technology for GEF (MVC).

GEF (MVC) - an interactive model-view-controller framework, which fosters the implementation of SWT-based tree editors and Draw2d-based graphical editors (and views) for the Eclipse UI Workbench.

Draw2d & GEF (MVC)

GEF 3.0 (2004)

- **Most** of the **features that make up GEF** where introduced:

  - **Rulers** & **Guides**, **Grid**

  - **Snap-to-Geometry**, Centered Resize, Match Size , Constraint Move and Resize, **Cloning**, **Panning**

  - Palette View, **Flyout Palette**, Palette Stacks

  - **Shortest Path** Connection **Routing**

- Since then, the **APIs** of Draw2d and GEF (MVC) **have stayed backwards compatible**.

But that's not all ...

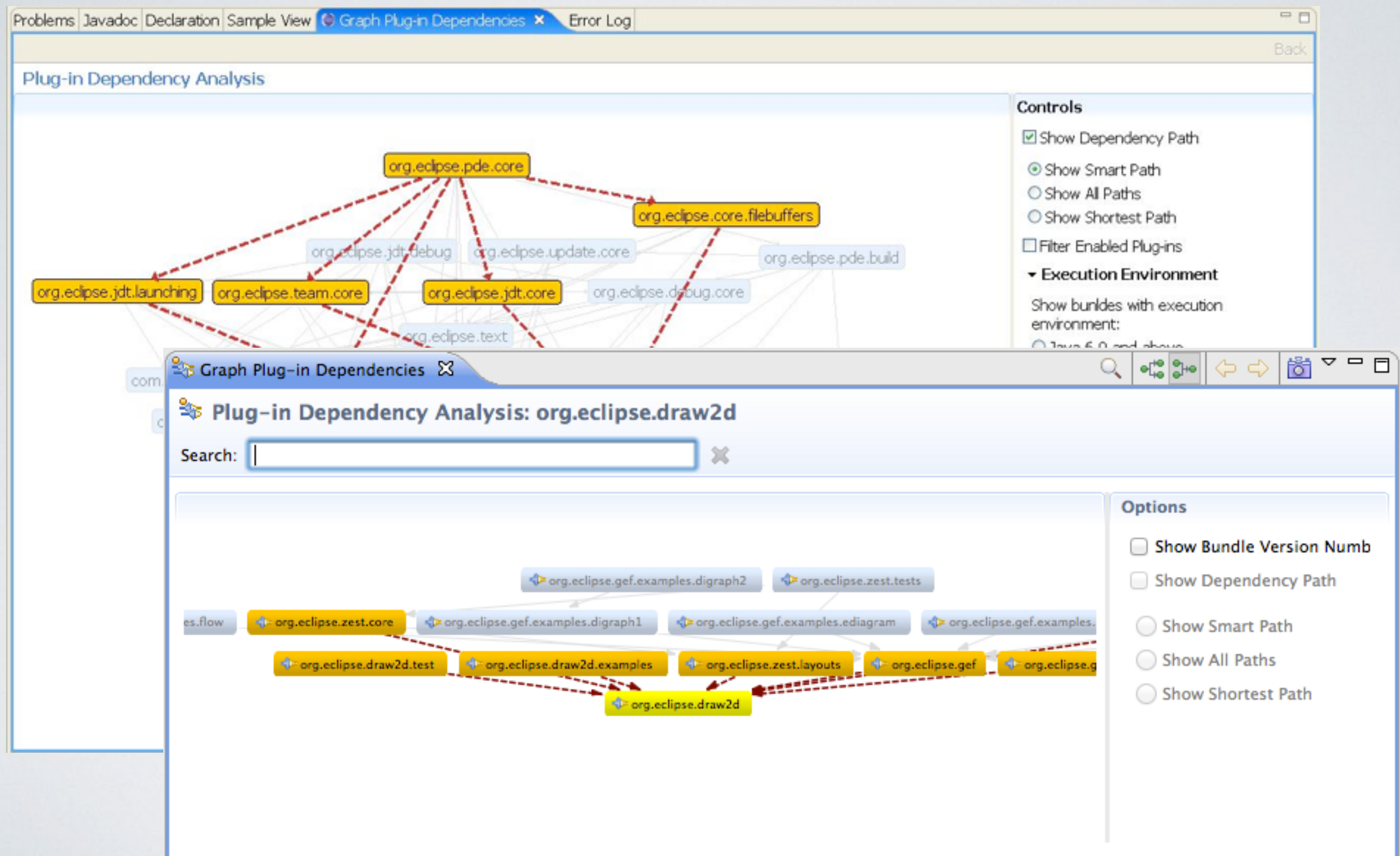Image: renjith krishnan / FreeDigitalPhotos.net

Zest

- Initial contribution of Zest by **University of Victoria** and **IBM Centre for Advanced Studies** as part of **Mylyn** in 2005.
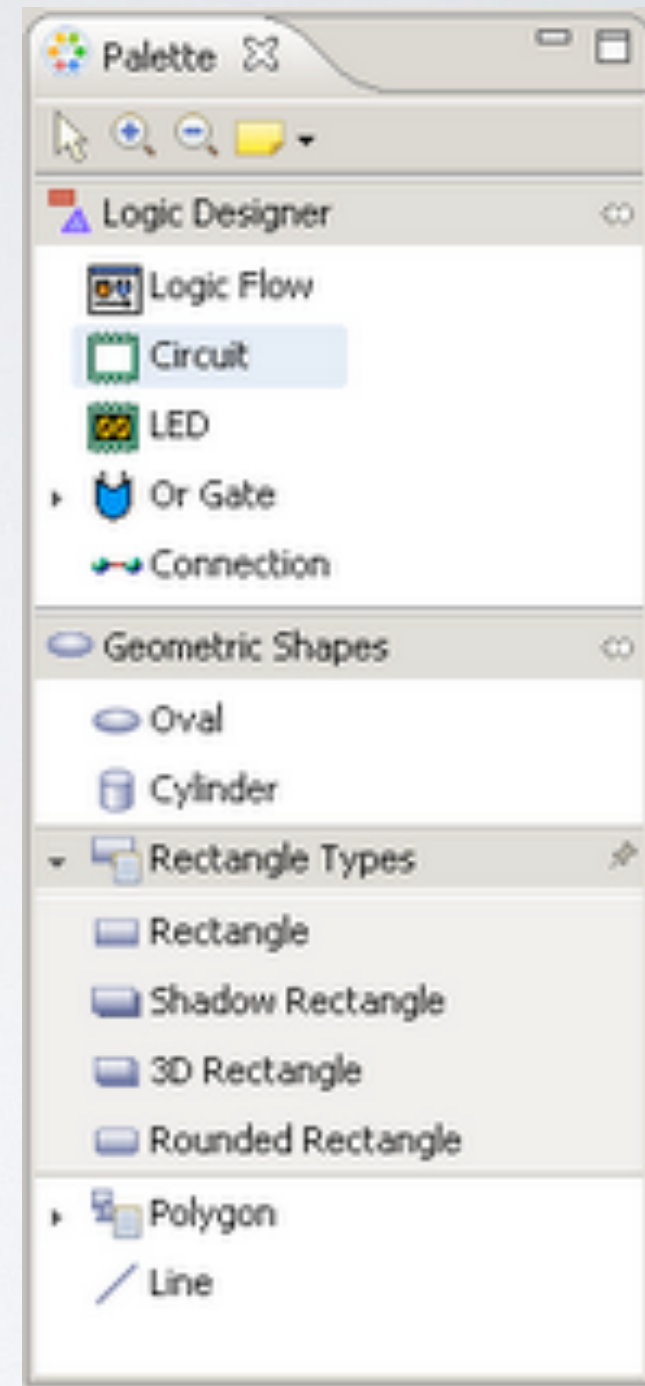- Joined in on GEF as **third component** with the **3.4 release** in **2007**.

Zest - a visualization toolkit based on SWT and Draw2d to support the implementation of views with automatic or semi-automatic layout for the Eclipse Workbench UI.
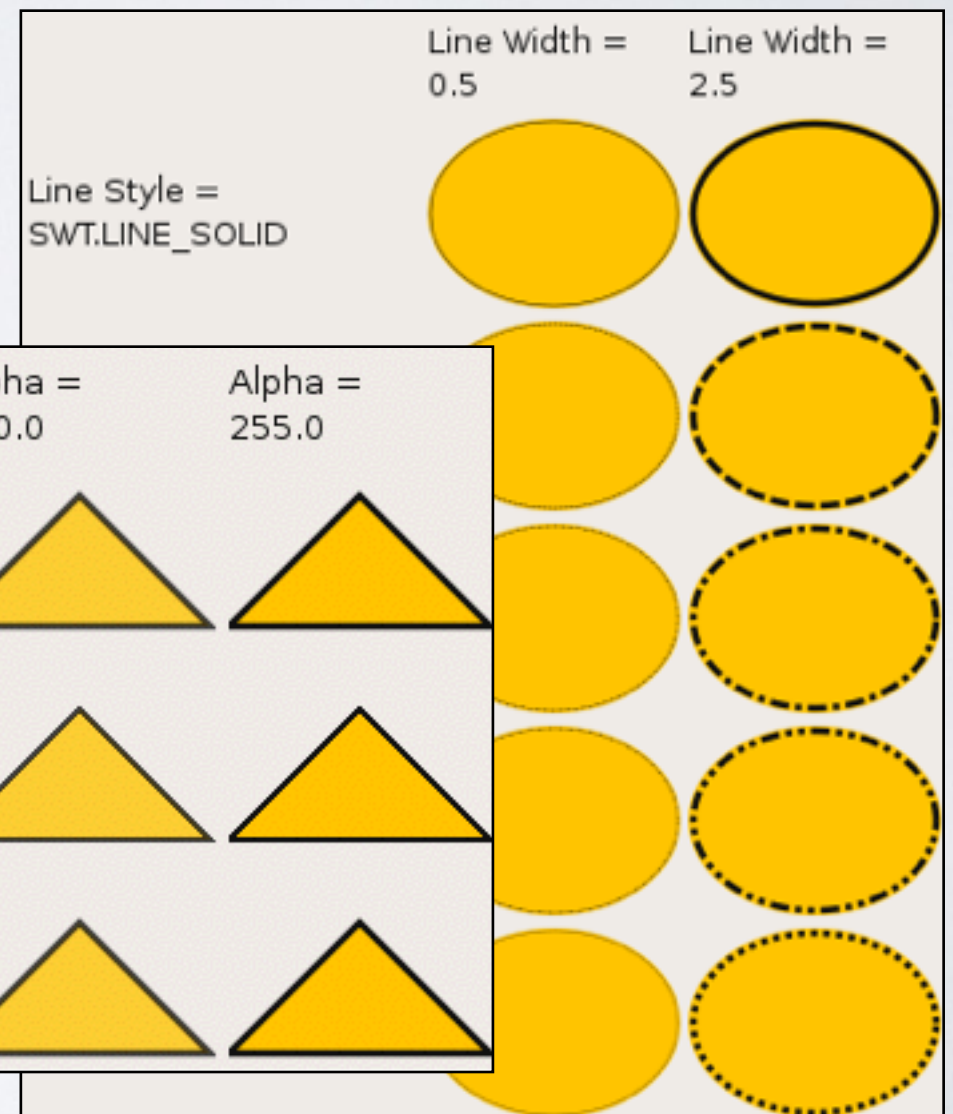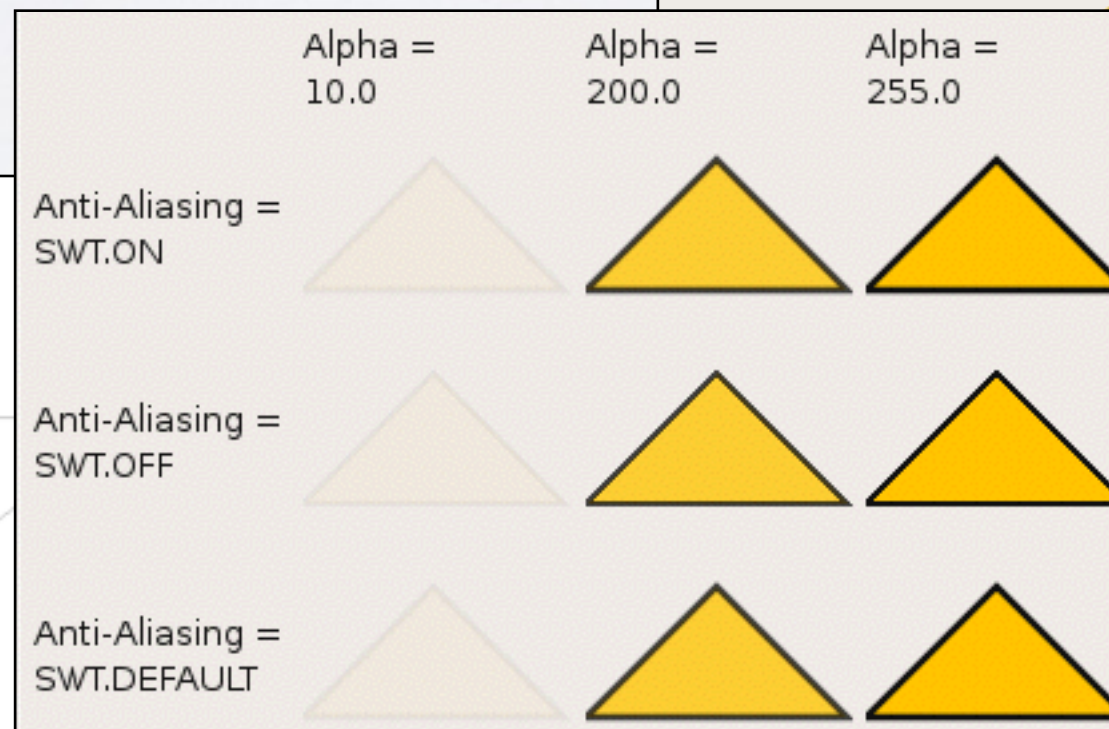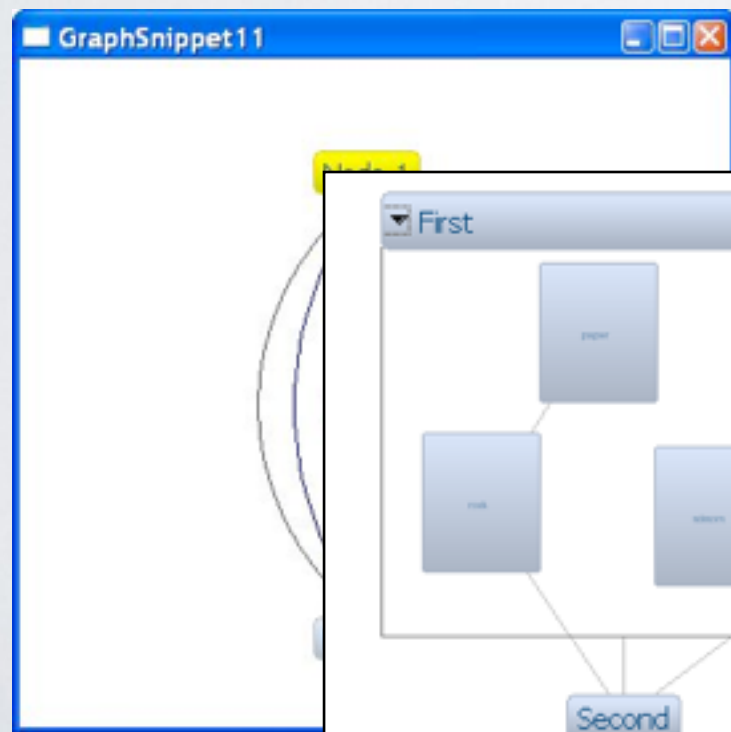
Zest

GEF 3.4 / Zest 1.0 (2008)

- **Modularization** into two components:
  - **Draw2d**
  - **GEF (MVC)**
- **Integration** of **Zest** as third component.
- GEF releng moved to **common modeling build** tools.
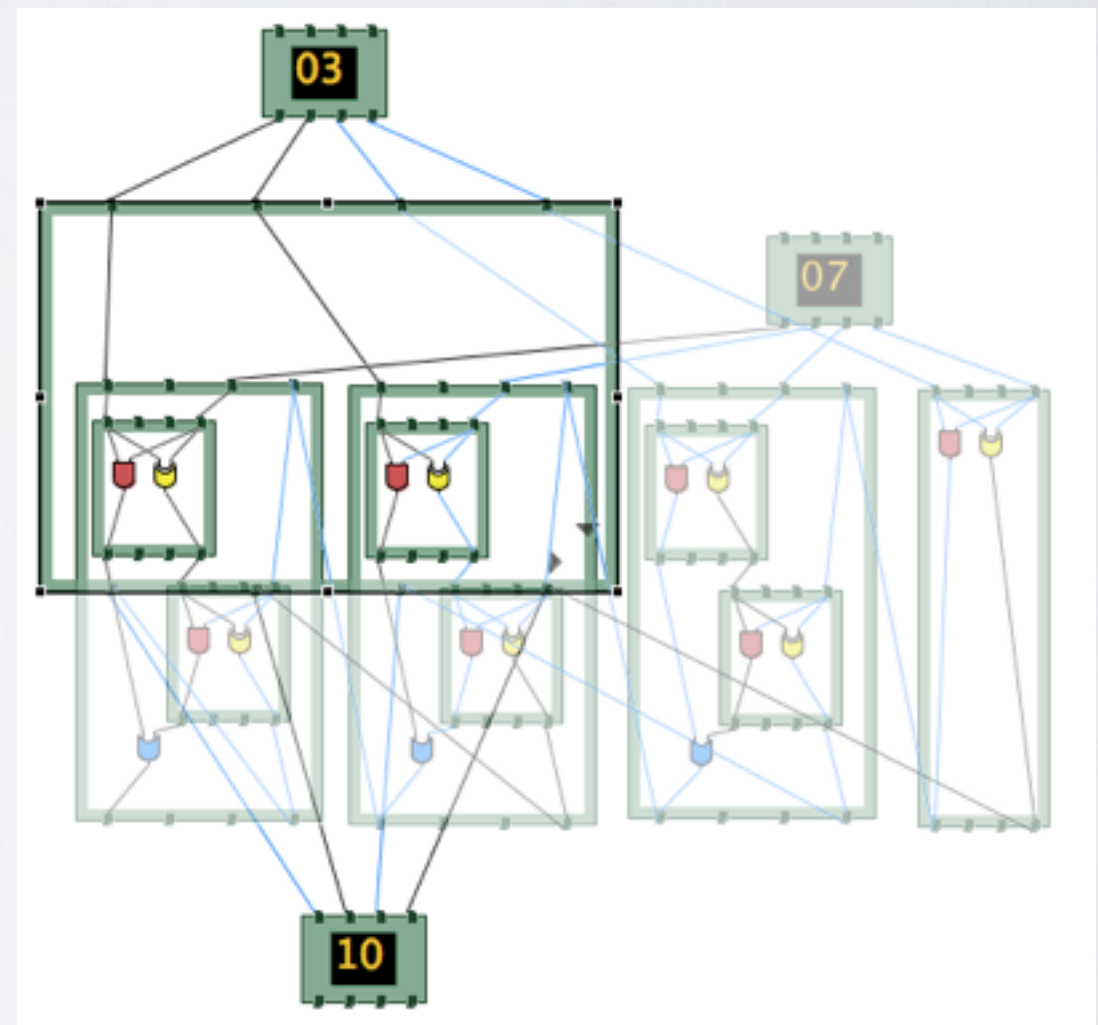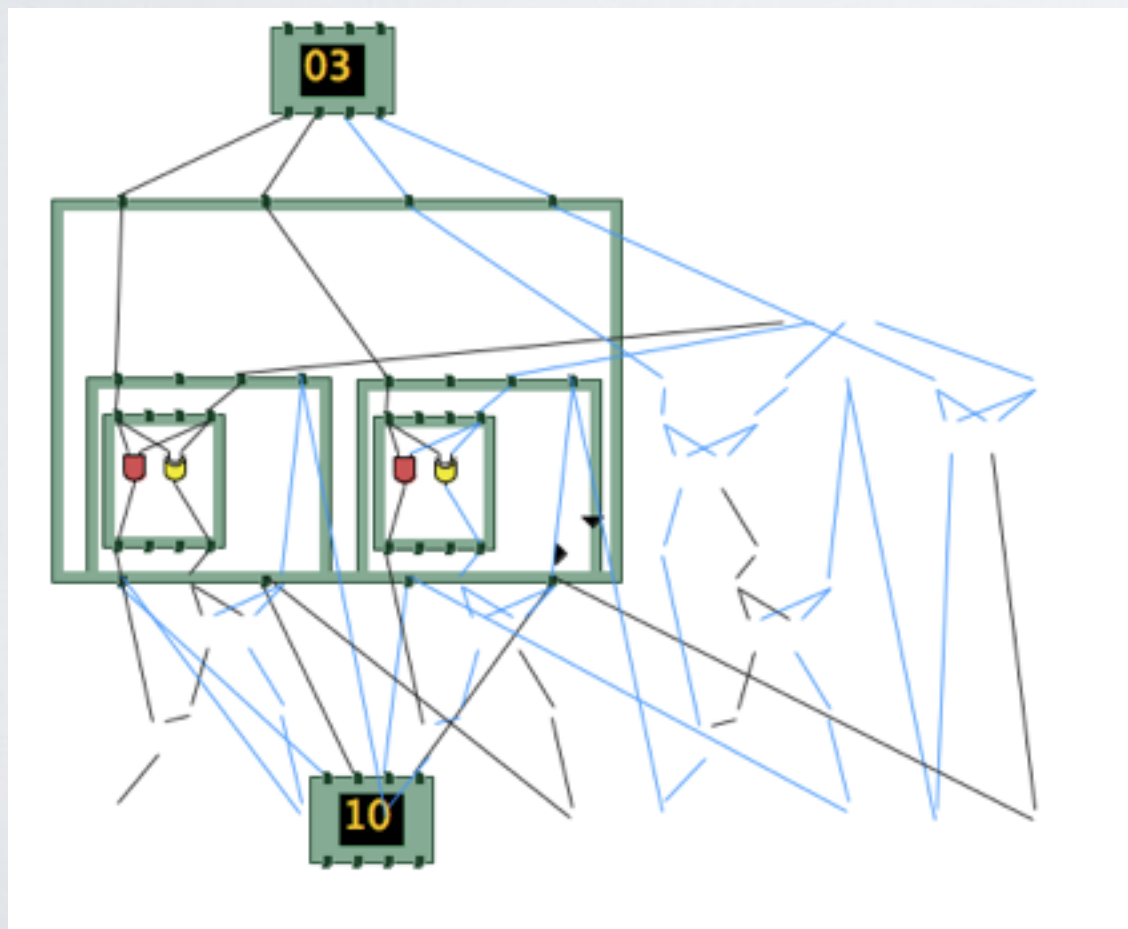- **Redesigned** UI for **Palette** (GEF)

GEF 3.5 / Zest 1.1 (2009)

- Support for **SWT line style attributes** (Draw2d)

- **Advanced Graphics** support (Draw2d)

- **Curved Connections** (Zest)

- **Nested content** provider (Zest)

GEF 3.6 / Zest 1.2 (2010)

- Improved clipping via **IClippingStrategy** (Draw2d)
- **Scrollable Feedback** support (GEF)

GEF 3.7 / Zest 1.3 (2011)

- Refactoring of Draw2d **geometry classes** (inconsistencies)

- Refactoring of **FlowLayout** and **ToolbarLayout** (introduced OrderedLayout)

- Refactoring of **LayoutEditPolicies**

- **Constraint Size-On-Drop** Creation

- Additional **Marquee Selection** strategies

- Refactoring of **UndoablePropertySheetPage** (undo/redo support while editor is inactive; opened for subclassing)

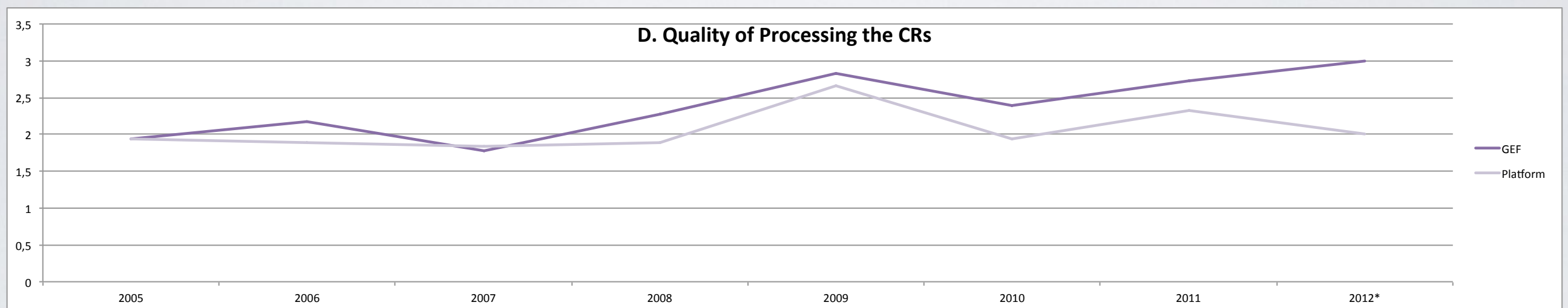# Present

GEF - Facts & Figures

- **10 committers** (*5 currently active,* 2 new in 2012)

  - Alex Boyko
  - *Anthony Hunter (Project Lead)*
  - *Alexander Nyßen*
  - Cherie Revells
  - *Fabian Steeg*

  - Ian Bull
  - Marc Gobeil
  - Nick Boldt
  - *Stephan Schwiebert*
  - *Zoltan Ujhelyi*

- **3 Git repositories** (gef, gef4, zest)

- **4 Hudson jobs** (gef-nightly-tycho, gef4-nightly-tycho, gef-zest-nightly, gef-zest-integration)

GEF 3.8 / Zest 1.4 (2012)

- **Migration** of source code repository from CVS **to Git**

- **Migration** of release engineering infrastructure from common modeling build **to Tycho/Hudson**:

  - gef-nightly-tycho build on **hudson.eclipse.org**

  - **nightly builds** as well as weekly **integration builds** are now **publicly available**

- **Bug fixes**, **clean-ups**, **no major new-and-noteworthy features**

GEF 3.8 / Zest 1.4

- Currently we have around **~400 open bugs/CRs**.

- Our **quality of processing** is **improving**...

- A lot of issues will not be resolvable without **breaking API**.

D. Quality of Processing the CRs

Metric calculated according to http://bugzillametrics.sourceforge.net/Process_Quality_Eclipse.pdf
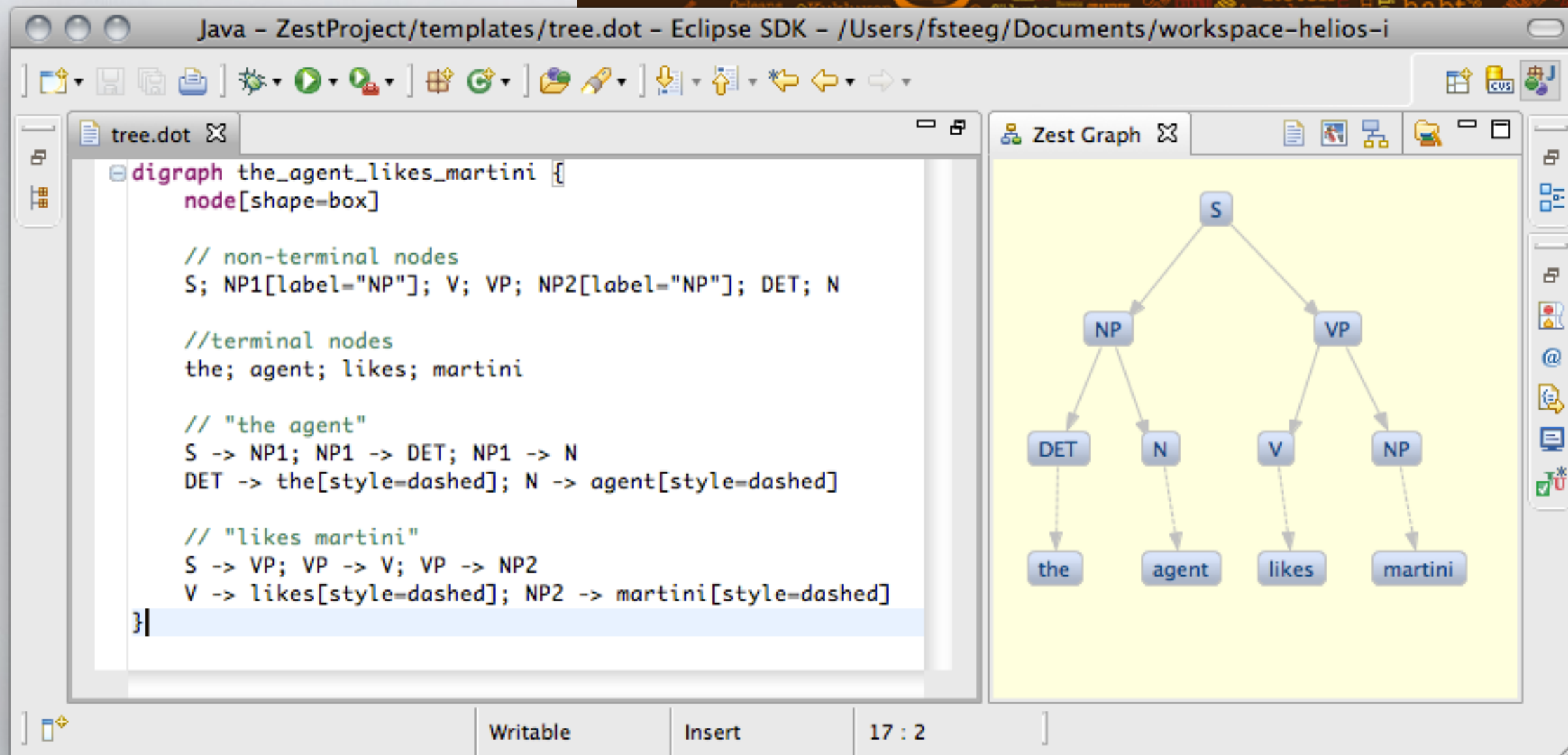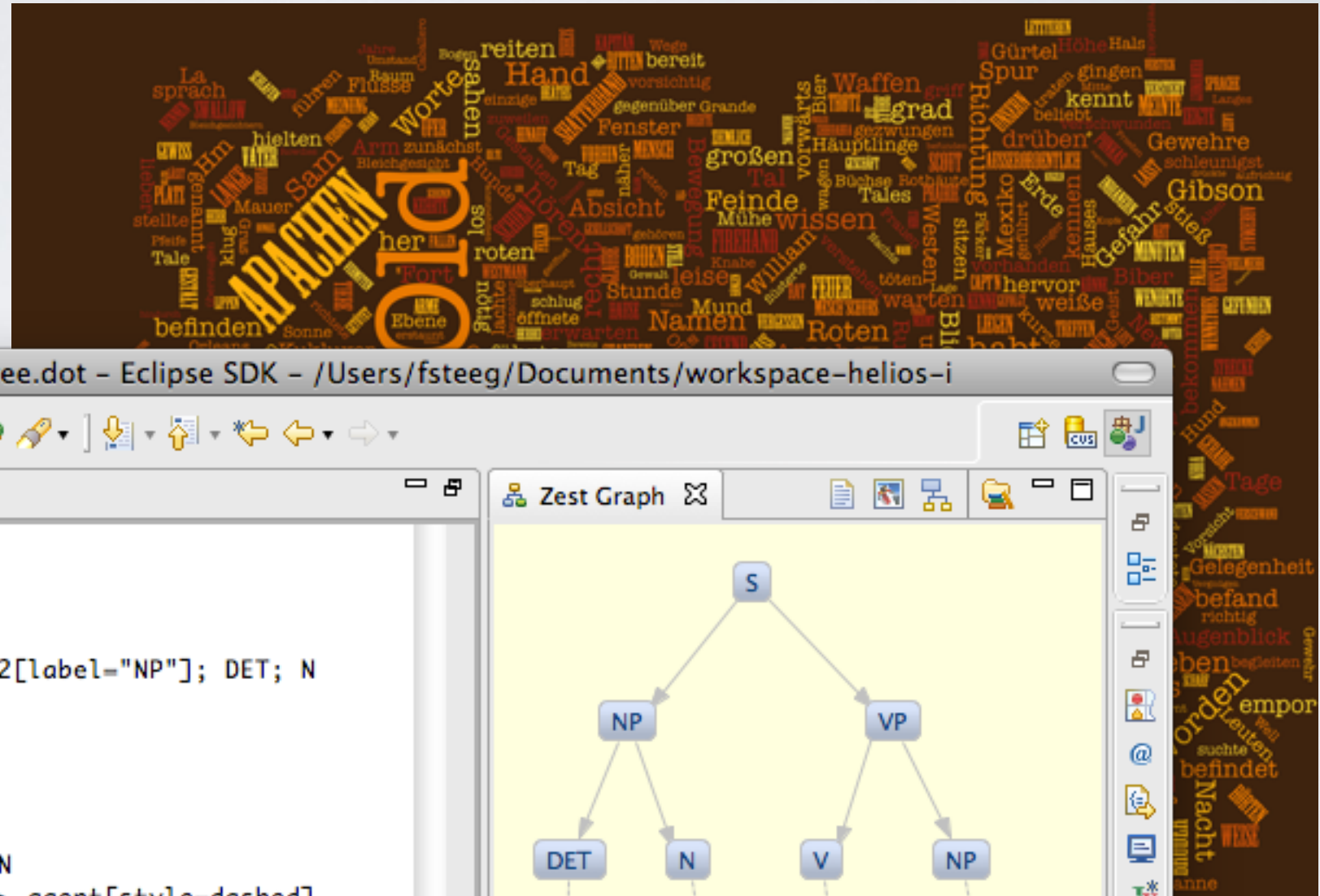
- How long does it take to fix a new CR?

- How often has a fixed CR to be reopened?

- How friction-less is the processing of CRs?

Zest 2 (since 2010)

- A provisional **Zest 2** component was initiated in 2010, to develop the **next generation Zest API**.

- Development takes place **in parallel to maintenance of Zest 1.x** in its own Git repository.

- **API** is regarded to be **provisional** up to the final graduation.

- Results are not provided as part of GEF 3.x / Zest 1.x but published via **Eclipse Marketplace**.

Zest 2 - New Features

- **Dot 4 Zest**

- **Cloudio**

GEF4 (since 2011)

- **GEF4** was initiated - according to Zest 2 - to **develop the next generation API** for Draw2d and GEF (MVC).

- Development is to take place **in parallel to maintenance** of **Draw2d / GEF (MVC) 3.x** in its own repository.

- Results are not provided provided as part of GEF 3.x / Zest 1.x but on **own update-sites**.

- Plan:

  - Create **new double-precision Geometry API** before 3.8.

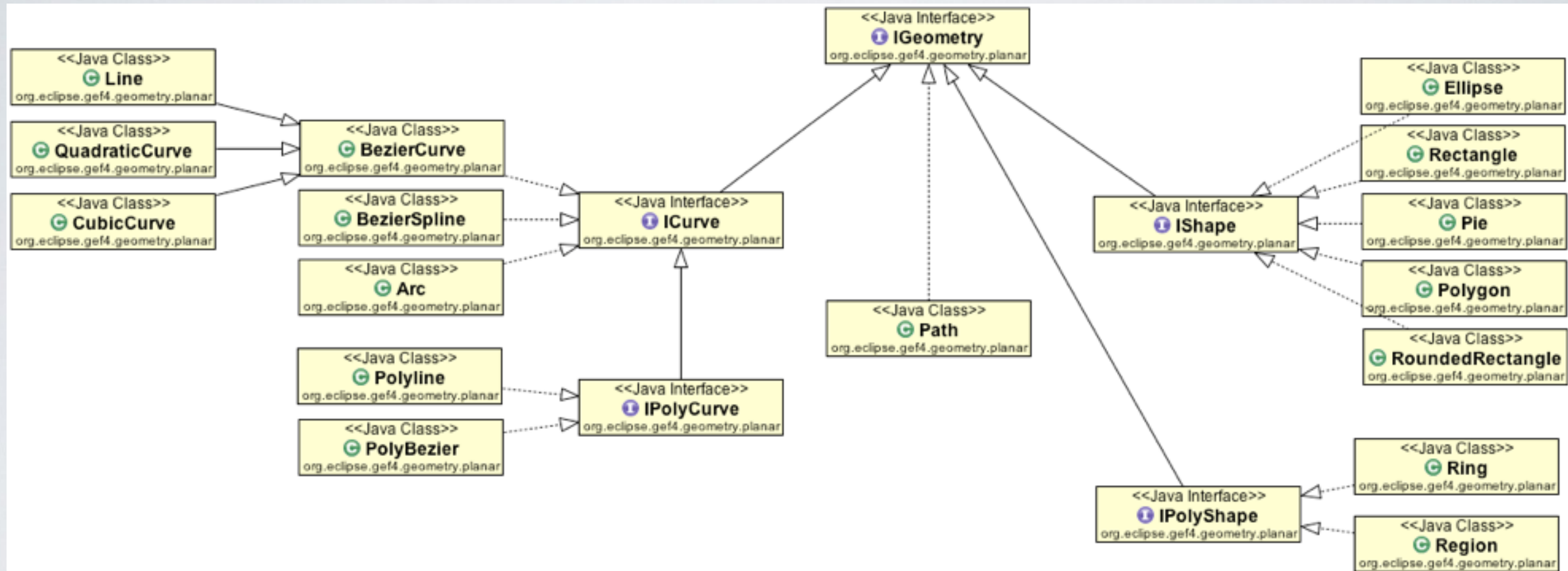 - **Migrate** the Draw2d and GEF (MVC) **3.8 code base** afterwards.

GEF4 Geometry*

- No low and high precision, just a single **double-precision API** (with **built-in imprecision** for comparisons).

- **Different geometric abstractions** for different purposes:

  - Euclidean (Vector, Straight)

  - Projective[+] (Vector3D[+], Straight3D[+])

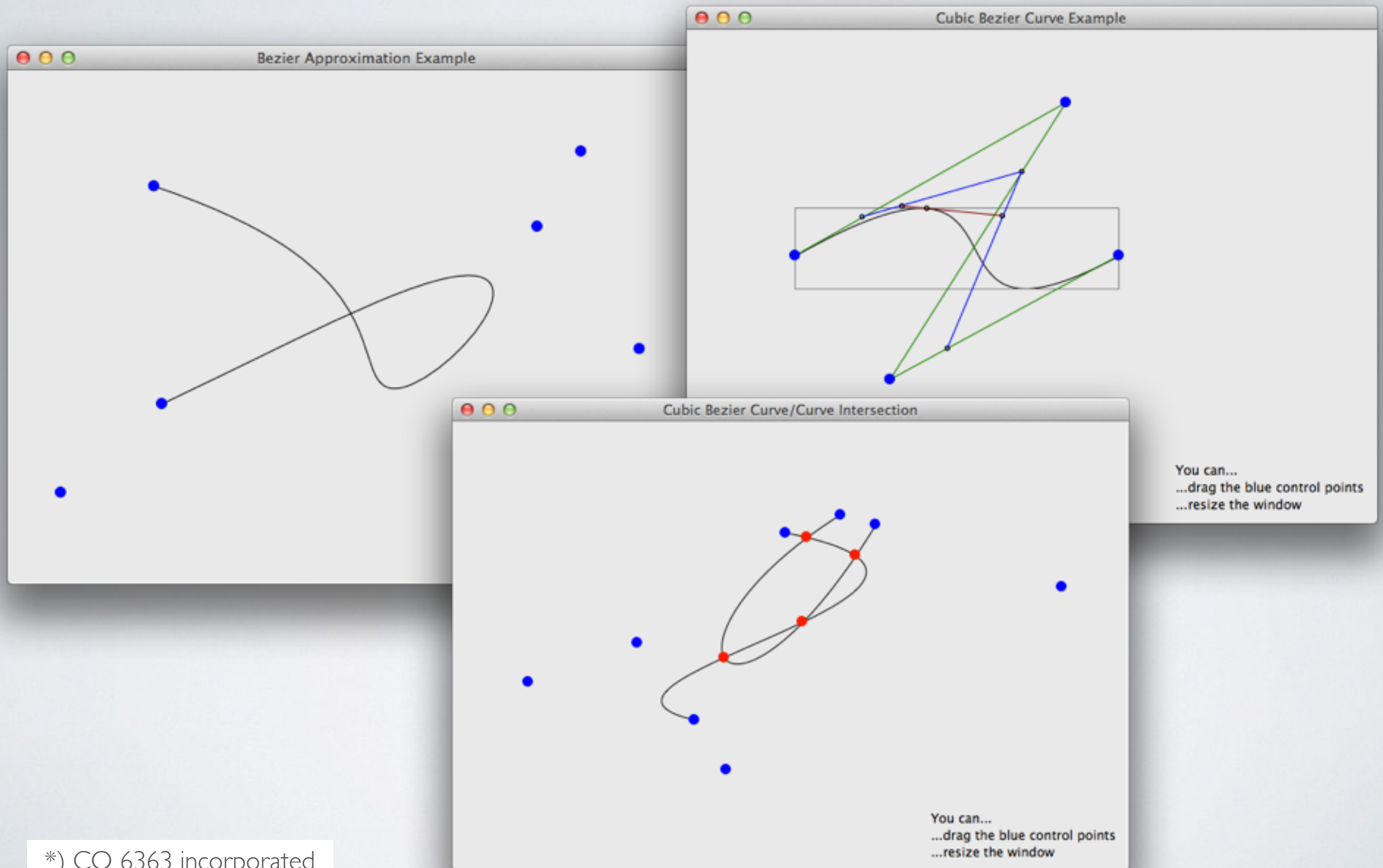 - Planar[+] (IGeometry[+], ICurve, IShape, IPolyCurve, IPolyShape)

GEF4 Planar Geometry*

GEF4 Geometry - Examples*

FUTURE

GEF4 + Zest 2 = GEF4

Unification of both approaches after Juno!

GEF4 - A Unified Approach

- A **unified approach** with a **shared code base** and a **common namespace** (org.eclipse.gef4) for all plug-ins.

- **Advantages:**

  - Clear **distinction** between **GEF proper** as the production component and **GEF4** as the provisional one

  - Chance to not only **refactor** GEF components but the **componentization** itself, which is only "historically" justified.

Some Topics for GEF4

- **Re-thinking** current **componentization**

- Support for the **E4 application model**

- Support for **rotation** and **other transformations**

- Support for **B-Spline-connections**

- Better **integrability** of native **SWT widgets**

- **Multi-touch gestures** support

- Revision of the **command framework**

- Revision of **connection handling** (clipping container)

- Various **renamings** and **restructurings** on the detail level...

Just one more thing...

- It's an often requested feature to use **GEF in the web**.

- This is **out of scope** for the GEF project itself, while **GEF4** might do **what is possible** to better **enable it**.

- Ports based on GEF 3.x / Zest 1.x:

  - RAP Incubator - GEF-Port

  - Architexa: http://www.codemaps.org/e/Eclipse_GEF

  - GEFGWT (GWT): http://gefgwt.org/

- There is an active discussion on gef-dev mailing list. Participate!

Thank You! Questions?