Graphiti
The Graphical Tooling Infrastructure
Speaking Plain Java

Michael Wenz
SAP AG

03/23/2010

Graphiti are:
Christian Brand
Jürgen Pasch
Matthias Gorning
Michael Wenz
Tim Kaiser
What Is Graphiti All About?

“The goal of Graphiti is to support the fast and easy creation of unified graphical tools, which can display and edit underlying domain models using a tool-defined graphical notation.”
Why Should I Want to Use Graphiti?

- Flat learning curve
  - Platform specific technology (e.g. GEF / Draw2D on Eclipse) is hidden from developer

- Fast and easy development of graphical editors for various domain models
  - First results with very low effort by using default implementations and strong framework

- Common look and feel with sensible defaults
  - User interactions were designed in close co-operation with usability specialists

- Option to support different platforms
  - Diagrams are defined platform independently
  - Currently only Eclipse is supported
  - Platform specifics (color-dialog, font-dialog, ...) can be integrated
How Does Graphiti Work?

- Screen
- Rendering Engine
- Interaction Component
- Pictogram Model
- Diagram Type Agent
- Domain Model
- Link Model

Graphiti Diagram Type Agent
What about the Diagram Type Agent?

A feature implements an operation for a domain object and defines the context in which the operation can be triggered.

Specialized features for common user interactions available.
Which Objects Do I Deal with?

**Domain**
- EClass
- EAttribute
- EOperation

**Result**
- StockItem
  - articleNr : Integer
  - name : String
  - description : String
  - quantity : String
  - doInventory() : void

**Links**
- Link

**Hierarchie Pictogram Elements**
- Container Shape
  - Shape
  - Container Shape
  - Shape
  - Container Shape
  - Shape

**Visualization Graphics Algorithms**
- Text
Well Then - How Do I Build a Tool?

Implement a Diagram Type Provider

Register the Diagram Type Provider for a Diagram Type

Implement a Feature Provider

Implement Features
Well Then - How Do I Build a Tool?

Implement a Diagram Type

Register the Diagram Type Provider for a Diagram Type

Implement a Feature Provider

Implement Features

```java
package org.eclipse.graphiti.tutorial;

import org.eclipse.graphiti.dt.AbstractDiagramTypeProvider;

public class TutorialDiagramTypeProvider extends AbstractDiagramTypeProvider {

    public TutorialDiagramTypeProvider() {
        super();
    }
}
```
Well Then - How Do I Build a Tool?

Implement a Diagram Type Provider

Register the Diagram Type Provider for a Diagram Type

Implement Features

...
Well Then - How Do I Build a Tool?

```java
package org.eclipse.graphiti.tutorial;

public class TutorialFeatureProvider extends DefaultFeatureProvider {

    public TutorialFeatureProvider(IDiagramTypeProvider dtp) {
        super(dtp);
    }
}
```

```
public TutorialDiagramTypeProvider() {
    super();
    setFeatureProvider(new TutorialFeatureProvider(this));
}
```
public class TutorialAddEClassFeature extends AbstractAddShapeFeature {

    public TutorialAddEClassFeature(IFeatureProvider fp) {
        super(fp);
    }

    @Override
    public boolean canAdd(IAddContext context) {
        // Check if user wants to add an EClass
        if (context.getNewObject() instanceof EClass) {
            // Check if user wants to add to a diagram
            if (context.getTargetContainer() instanceof Diagram) {
                return true;
            }
        }
        return false;
    }

    @Override
    public PictogramElement add(IAddContext context) {
        ...
    }
}
Well Then - How Do I Build a Tool?

```java
public PictogramElement add(IAddContext context) {
    EClass addedClass = (EClass) context.getNewObject();
    Diagram targetDiagram = (Diagram) context.getTargetContainer();
    int w = context.getWidth() <= 0 ? 100 : context.getWidth();
    int h = context.getHeight() <= 0 ? 50 : context.getHeight();

    ContainerShape containerShape = PeUtil.createContainerShape(targetDiagram, true);
    RoundedRectangle roundedRectangle = GaUtil.createRoundedRectangle(containerShape, 5, 5);
    roundedRectangle.setForeground(manageColor(CLASS_FOREGROUND));
    roundedRectangle.setBackground(manageColor(CLASS_BACKGROUND));
    roundedRectangle.setLineWidth(2);
    GaUtil.setLocationAndSizeOfGraphicsAlgorithm(roundedRectangle, context.getX(), context.getY(), w, h);
    link(containerShape, addedClass);

    Shape shape = PeUtil.createShape(containerShape, false);
    Text text = GaUtil.createDefaultText(shape, addedClass.getName());
    text.setForeground(manageColor(CLASS_TEXT_FOREGROUND));
    text.setHorizontalAlignment(Orientation.ALIGNMENT_CENTER);
    text.setVerticalAlignment(Orientation.ALIGNMENT_CENTER);
    text.getFont().setBold(true);
    GaUtil.setLocationAndSizeOfGraphicsAlgorithm(text, 0, 0, width, 20);

    shape = PeUtil.createShape(containerShape, false);
    Polyline polyline = GaUtil.createPolyline(shape, new int[] { 0, 20, width, 20 });
    polyline.setForeground(manageColor(CLASS_FOREGROUND));
    polyline.setLineWidth(2);

    return containerShape;
}
```
Well Then - How Do I Build a Tool?

Implement a Diagram Type

Implement a Feature Provider

Implement Features
Graphiti – What else to Mention?