Graphical Editing Framework

Randy Hudson
Pratik Shah
IBM Rational Software
Agenda

- Introduction
- Shortest Path Connection Routing
- New EMF Diagram example
- WYSIWYG Text Editing
- Get Involved
- Q&A
Introduction

- Demo of GEF-based applications
The Shortest Path Problem

- Maintaining manual bendpoints is a burden on the User
- Requirements:
  - Connections should automatically avoid nodes
  - Must perform well, especially interactively
  - Some nodes cannot be avoided
  - Must support manual bendpoints
- Approach:
  - Build on existing research for single-path cases
  - Reduced visibility graph
  - Naïve implementation: $O(n^3)$
  - Dijkstra’s algorithm
Shortest Path Routing
Reduced Visibility Graph
Incremental Generation

1) Try to add a segment
2) Hit an unknown obstacle, stop and queue that obstacle
3) Hit known obstacle, throw out segment

Queued: 1
Processed:
Incremental Generation

4) Generate segments for next queued obstacle
5) Repeat with each segment

Queued: 2, 3
Processed: 1
Incremental Generation

Queued: 3, 4
Processed: 1, 2
Be Optimistic

Queued: 3, 4
Processed: 1, 2
Optimistic/Incremental Gains
Optimistic Visibility Graph
Find Path Using Dijkstra’s

- If end is not reachable, relax or remove Oval
Routing multiple paths

- Paths appear to converge or cross
- Offset each path a different amount
- Isn’t order important?
- Are offset segments still “visible”?
Offset connections

- Offsetting causes new intersections
- Proper ordering can avoid unnecessary path crossing
Intersections Introduced by Offsetting

- Test *fuzzy* segments against other *deformed* obstacles
Routing Multiple Paths

- Divide segments by inserting the intersecting vertices
- Update the obstacles and segments
Augmented Shortest Path

- Repeat a few times
- Or until no more changes
Determining Path Order for Offsets

- Walk each path start to end
- Label vertices IN or OUT
- Do any paths you touched next
- Resolve labeling conflicts
Topologically Sort all Paths

- Visit each vertex of each path
- Compare angles of remaining paths
- If angle >=, path comes later
- If <, path must come before current
- Offset IN inside-out, OUT outside-in
Incremental Updating

✓ Yes, we have that
EDiagram Example

- **Goals:**
  - Demonstrate “business” and “view” model separation
  - Experience EMF+GEF integration challenges first-hand
  - Identify new function missing from GEF
  - Provide community with graphical, EPL, ecore schema editor

- `org.eclipse.gef.examples.ediagram`
  - ~50 classes + generated model
  - Requires EMF 2.0
  - Load from CVS
Business and View Model Separation

- my.ediagram
  - View Model
  - A graphical diagram
  - Imports 1 or more ecore files
  - Contains views of elements in the schema
  - Graphical information

- myschema.ecore
  - Business Model
  - The schema
  - Does not refer to the diagram
  - Defines packages, classes, etc.
  - Semantic information
Example Features

- Outline displays business model
- Create new items using the Palette
  OR
- DND from outline view
- Graphically edit inheritance/references
- Sticky Notes for documentation
- Properties displayed in PropertySheet
**EDiagram Example**

- **Results**
  - 6 weeks to learn EMF and build initial implementation
  - Still needs work before it can replace existing editors

- **To-Dos**
  - Import files instead of packages?
  - Invoke EMF utilities such like “Validate Model”
  - Improve Property Sheet Support
  - Enhance editing of attributes/operations
  - More actions/context menus
  - Finer control of visual aspects of diagram
WYSIWYG Text Editing

- Requirements
  - Display “rich” text documents
  - Still no restriction on model
  - Support Bidi, DBCS, etc.

- What’s different
  - Model is hierarchical/structured (DOM)
  - Supports mixed font families and sizes
  - Document may contain non-textual elements
  - “Rich” formatting includes padding, borders, bullets, images, etc.
org.eclipse.draw2d.text

- Base package for rich text function
- Shipped in 2.1 for GEF palette
- Is being enhanced
- 2 types of special-purpose figures
  - **Block** figure
    - Rectangular in form
    - Creates and manages “lines”
  - **Inline** figure
    - Contains ≥1 rectangular *fragments*
    - May span ≥1 lines
- Special layout requirements
Example: Represent Simple HTML

Source:

```html
<p>Returns the active page, or <code>null</code> if there is none</p>
```

DOM:

```
Root
  |- Text
  |- Node
    `- Text
      `- Text
```

Figures:

```
BlockFlow
  |- TextFlow
  `- InlineFlow
       `- TextFlow
```

Graphical Editing Framework | © 2005 by International Business Machines; made available under the EPL v1.0
WYSIWYG Editing

- Caret management
  - Navigation using keyboard: HOME, PAGE_UP, etc.
  - Placement using the mouse
  - Swipe-selection, double-click, etc.
- Editing the document
  - Actions affect current selection
  - Ask editpart for command
- Special concerns
  - What does CTRL+B do if selection is empty?
  - What should be selected on undo/redo?
Get Involved!

Contributors to the GEF Project
(chronologically)

- Bug reporters
- Alex Selkov
- Gunnar Wagenknecht
- Elias Volanakis
- Régis Lemaigre
- Asim Ullah
- Brian Fernandes
- Bo Majewski
- Phil Zoio

Ways **YOU** can contribute:

- Submit patches
- New function
- Documentation
- Write an Article
- Monitor Newsgroup
- Website construction
- 2 Words: JUNIT
- Performance Tests
- Platform Testing
- @see helpwanted
Questions