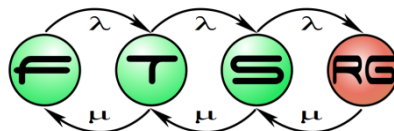


Grafikus szerkesztők

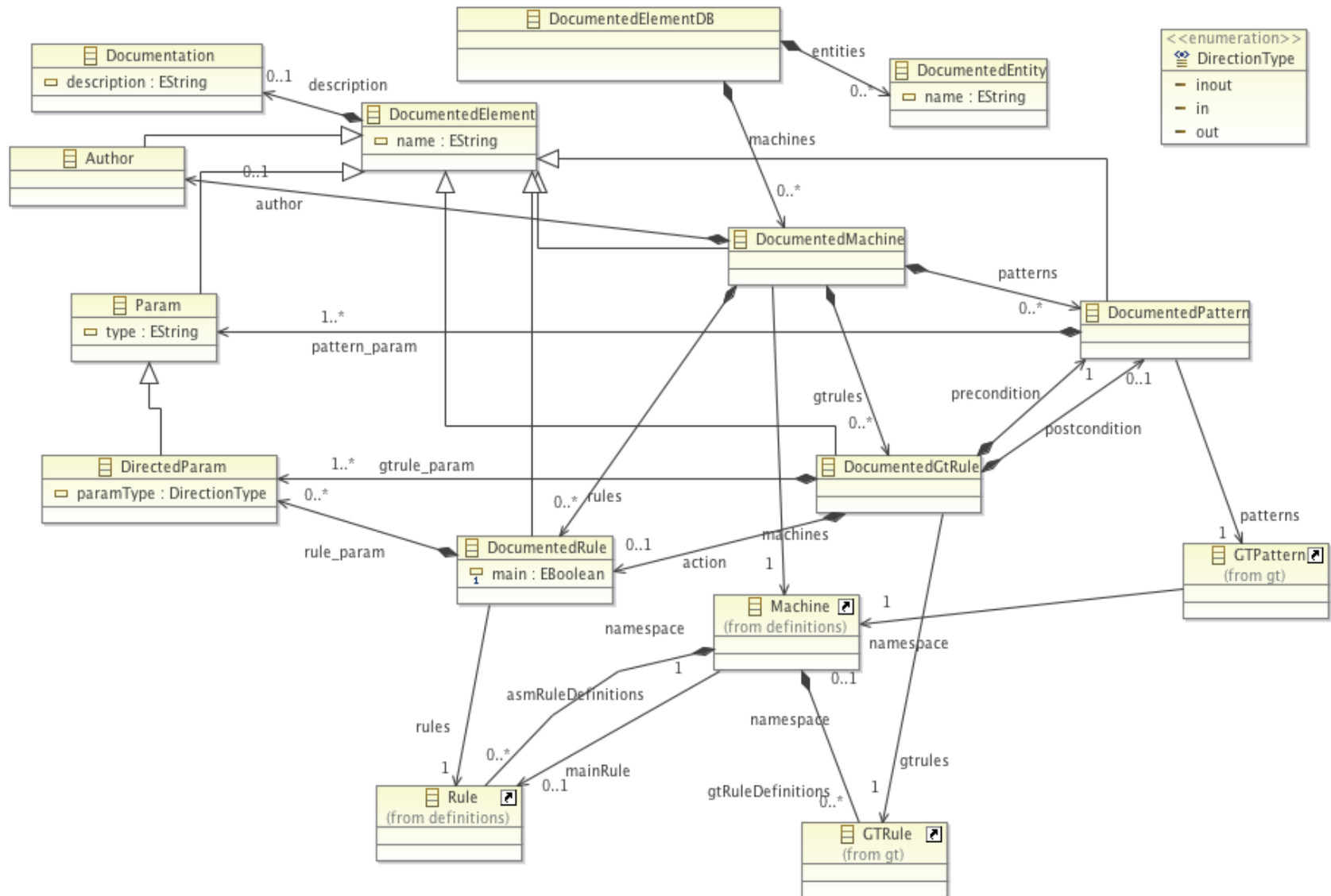
Kiegészítések
Layouting, 3D



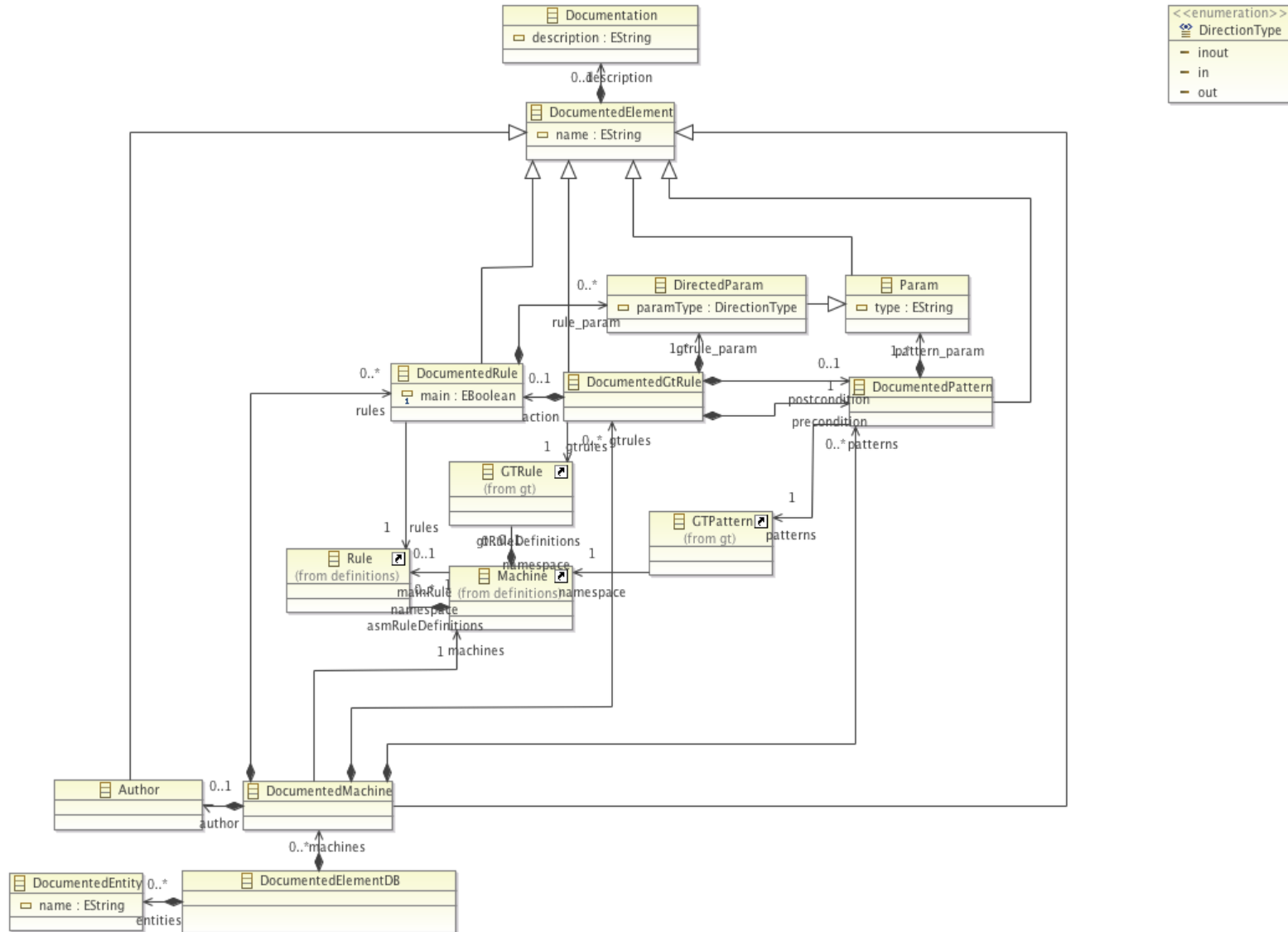
Grafikus szerkesztők

- Modellek megjelenítése
 - (Gráf szerkezet)
 - Tipikusan
 - Lassú szerkesztés
 - Jó áttekinthetőség

Átlátható?



Így jobb...



Gráfok elrendezése

Gráf rajzolás

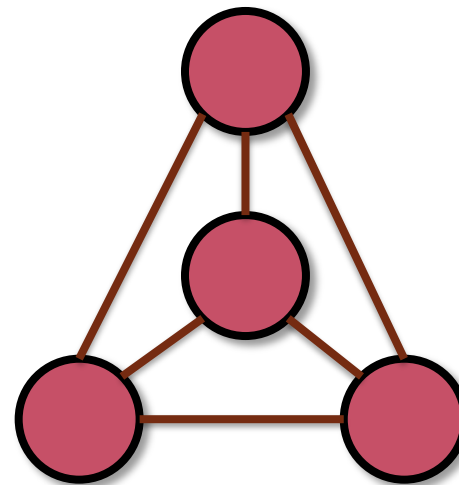
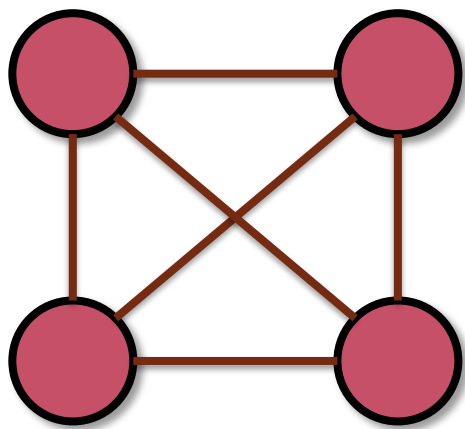
- Matematika

- $G=(V,E)$
- Kiolvasva: csomópontok és élek
- Nincs megjelenítési információ

- Gráf rajzolás

- Rendeljük megjelenítési információt a gráf csomópontokhoz!

Hogyan rajzoljunk?



- Nincs egységes követelményrendszer
 - Mit akarunk hangsúlyozni?

Gráfrajzoló eszközök

- GraphViz
 - Nem Eclipse-es
- Zest
 - Akadémiai fejlesztés
 - GEF alprojekt
- KIELER
 - Akadémiai fejlesztés
 - Eclipse alapú
- ...

Zest: The Eclipse Visualization Toolkit

- Egyszerű gráfrajzoló könyvtár
 - SWT widget
 - Adat megadása
 - Layout algoritmus megadása

Zest: SWT jellegű felhasználás

- Graph widget
 - addNode
 - addConnection
 - add

Zest példa

```
Graph g = new Graph(shell, SWT.NONE);
```

```
GraphNode n = new GraphNode(g, SWT.NONE, "Paper");
```

```
GraphNode n2 = new GraphNode(g, SWT.NONE, "Rock");
```

```
GraphNode n3 = new GraphNode(g, SWT.NONE, "Scissors");
```

```
new GraphConnection(g, SWT.NONE, n, n2);
```

```
new GraphConnection(g, SWT.NONE, n2, n3);
```

```
new GraphConnection(g, SWT.NONE, n3, n);
```

```
g.setLayoutAlgorithm(new SpringLayoutAlgorithm  
    (LayoutStyles.NO_LAYOUT_NODE_RESIZING), true);
```

Zest példa

```
Graph g = new
```

```
GraphNode n1
```

```
GraphNode n2
```

```
GraphNode n3
```

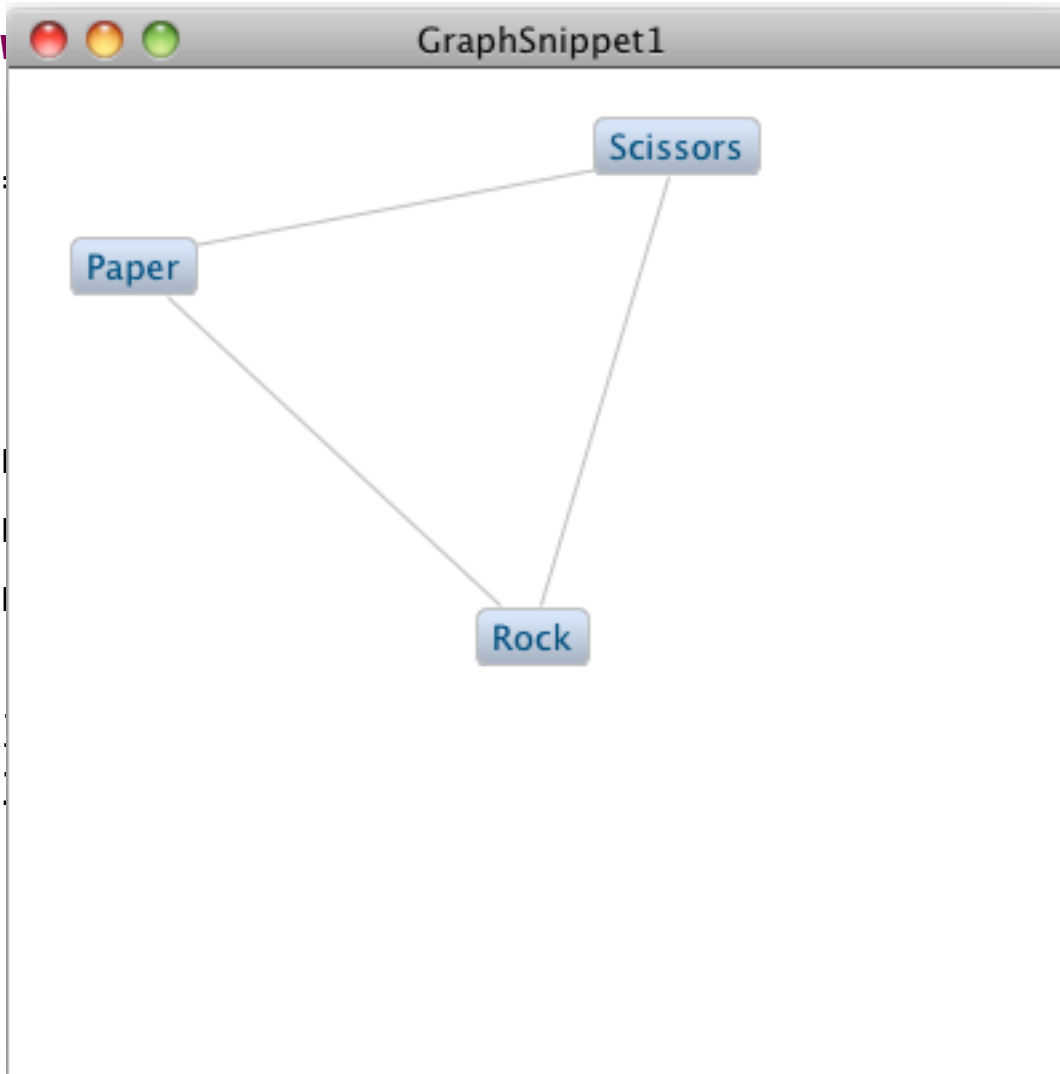
```
new GraphCon
```

```
new GraphCon
```

```
new GraphCon
```

```
g.setLayoutA
```

```
(LayoutSty
```



;

Zest: JFace API

- GraphViewer osztály
 - JFace viewer megszokott funkcionalitás
 - Content Provider
 - Label Provider
 - Selection API
 - Filter, Ordering
 - ...

Content Provider

- Adat feltöltéshez három forrás támogatott
 - IGraphContentProvider
 - getElements(): élek listája
 - getSource(rel), getTarget(rel): kezdő- és végpont
 - IGraphEntityContentProvider
 - getElements(): csomópontok listája
 - getConnectedTo(entity): szomszéd csomópontok
 - IGraphEntityRelationshipContentProvider
 - getElements(): csomópontok listája
 - getRelationships(entity, entity): élek két csomópont között

Content Provider példa

```
class MyContentProvider implements IGraphEntityContentProvider {
    public Object[] getElements(Object inputElement) {
        return new String[] { "First", "Second", "Third" };
    }
    public Object[] getConnectedTo(Object entity) {
        if (entity.equals("First")) { return new Object[] { "Second" }; }
        if (entity.equals("Second")) { return new Object[] { "Third" }; }
        if (entity.equals("Third")) { return new Object[] { "First" }; }
        return null;
    }
    public double getWeight(Object entity1, Object entity2) {
        return 0;
    }
    public void dispose() {}
    public void inputChanged(Viewer viewer, Object oldInput,
        Object newInput) {}
}
```

Content Provider példa

```
class MyContentProvider implements IGraphEntityContentProvider {  
    public Object[] getElements(Object inputElement) {  
        return new String[] { "First", "Second", "Third" };  
    }  
    public Object[] getConnectedTo(Object entity) {  
        if (entity.equals("First")) { return new Object[] { "Second" }; }  
        if (entity.equals("Second")) { return new Object[] { "Third" }; }  
        if (entity.equals("Third")) { return new Object[] { "First" }; }  
        return null;  
    }  
    public double getWeight(Object entity1, Object entity2) {  
        return 0;  
    }  
    public void dispose() {}  
    public void inputChanged(Viewer viewer, Object oldInput,  
        Object newInput) {}  
}
```

Csomópont
objektumok

Content Provider példa

```
class MyContentProvider implements IGraphEntityContentProvider {  
    public Object[] getElements(Object inputElement) {  
        return new String[] { "First", "Second", "Third" };  
    }  
    public Object[] getConnectedTo(Object entity) {  
        if (entity.equals("First")) { return new Object[] { "Second" }; }  
        if (entity.equals("Second")) { return new Object[] { "Third" }; }  
        if (entity.equals("Third")) { return new Object[] { "First" }; }  
        return null;  
    }  
    public double getWeight(Object entity1, Object entity2) {  
        return 0;  
    }  
    public void dispose() {}  
    public void inputChanged(Viewer viewer, Object oldInput,  
        Object newInput) {}  
}
```

Szomszéd
objektumok

Content Provider példa

```
class MyContentProvider implements IGraphEntityContentProvider {  
    public Object[] getElements(Object inputElement) {  
        return new String[] { "First", "Second", "Third" };  
    }  
    public Object[] getConnectedTo(Object entity) {  
        if (entity.equals("First")) { return new Object[] { "Second" }; }  
        if (entity.equals("Second")) { return new Object[] { "Third" }; }  
        if (entity.equals("Third")) { return new Object[] { "First" }; }  
        return null;  
    }  
    public double getWeight(Object entity1, Object entity2) {  
        return 0;  
    }  
    public void dispose() {}  
    public void inputChanged(Viewer viewer, Object oldInput,  
        Object newInput) {}  
}
```

Élsúlyok

Content Provider példa

```
class MyContentProvider implements IGraphEntityContentProvider {  
    public Object[] getElements(Object inputElement) {  
        return new String[] { "First", "Second", "Third" };  
    }  
    public Object[] getConnectedTo(Object entity) {  
        if (entity.equals("First")) { return new Object[] { "Second" }; }  
        if (entity.equals("Second")) { return new Object[] { "Third" }; }  
        if (entity.equals("Third")) { return new Object[] { "First" }; }  
        return null;  
    }  
    public double getWeight(Object entity1, Object entity2) {  
        return 0;  
    }  
    public void dispose() {}  
    public void inputChanged(Viewer viewer, Object oldInput,  
        Object newInput) {}  
}
```

Inputkezelés

Label Provider

- Egyszerű JFace Label provider
 - Objektumokhoz szöveg/kép rendelés
- Kiegészítő lehetőségek
 - IEntityStyleProvider
 - IConnectionStyleProvider
 - IFigureProvider

Label Provider példa

```
class MyLabelProvider extends LabelProvider {
    final Image image = Display.getDefault().getSystemImage
(SWT.ICON_WARNING);

    public Image getImage(Object element) {
        if (element instanceof String) { return image; }
        return null;
    }

    public String getText(Object element) {
        if (element instanceof String) {
            return element.toString();
        }
        return null;
    }
}
```

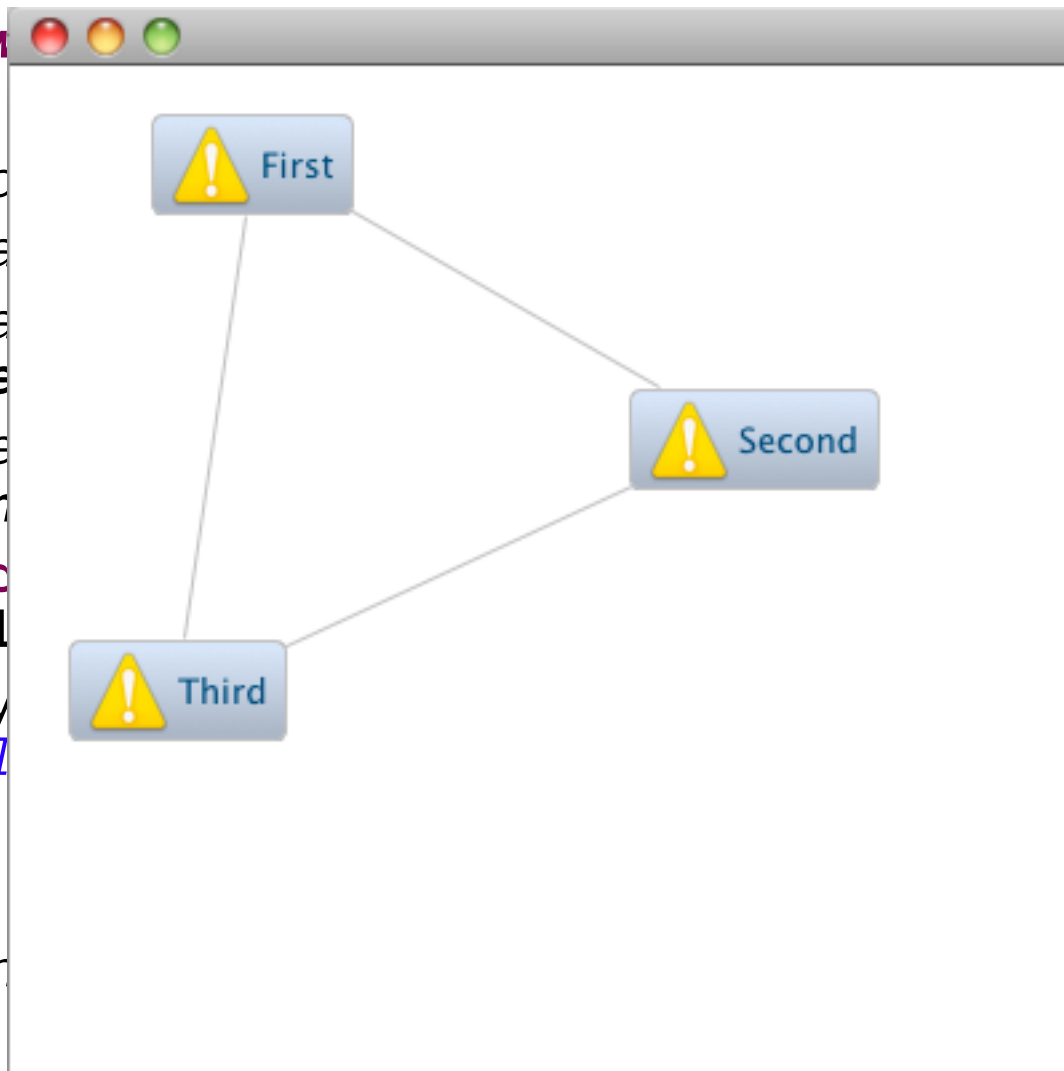
Zest: JFace API példa

```
viewer = new GraphViewer(shell, SWT.NONE);

viewer.setContentProvider(new MyContentProvider());
viewer.setLabelProvider(new MyLabelProvider());
viewer.setLayoutAlgorithm(new SpringLayoutAlgorithm
(LayoutStyles.NO_LAYOUT_NODE_RESIZING));
viewer.addSelectionChangedListener(new
ISelectionChangedListener() {
    public void selectionChanged
        (SelectionChangedEvent event) {
        System.out.println
        ("Selection changed: " + (event.getSelection()));
    }
});
viewer.setInput(new Object());
```

Zest: JFace API példa

```
viewer = new  
  
viewer.setCo  
viewer.setLa  
viewer.setLa  
(LayoutStyle  
viewer.addSe  
ISelectionCh  
  
public  
    (Sel  
        Sy  
        ("Sel  
    }  
});  
viewer.setIn
```



```
er());  
);  
orithm
```

```
ection()));
```

Elrendezések

- Gráfelrendezések megadása
 - Rendező algoritmus (layout algorithm)
 - Sokféle beépítve
 - Grid
 - Radial, Tree
 - Spring
 - Bővíthető
 - Meglevő algoritmus paraméterezésével
 - Több algoritmus egymás után futtatásával
 - Teljesen saját algoritmus írásával

Zest és GEF/GMF

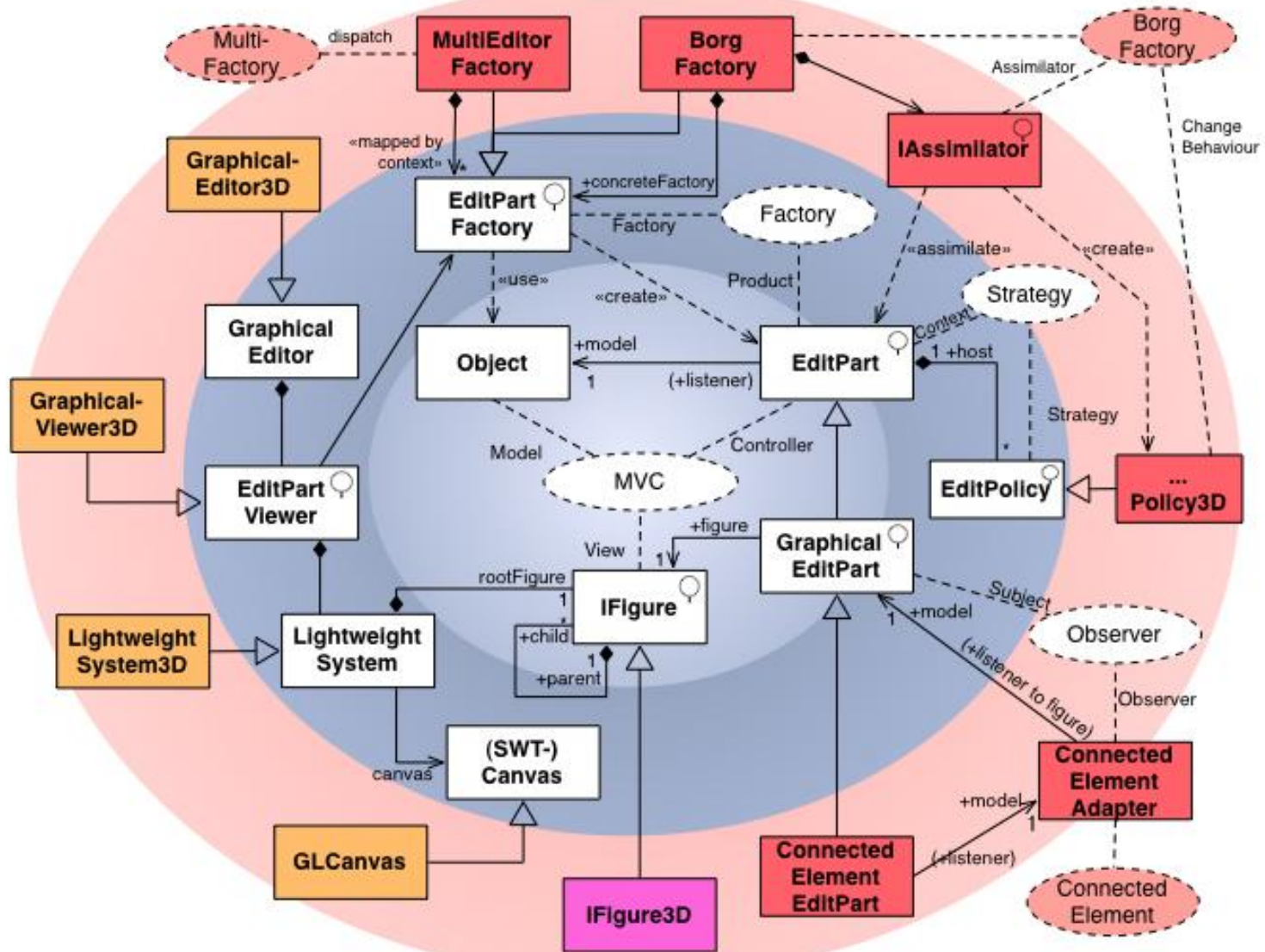
- Hogy kapcsolódik a Zest a GEF-hez?
 - Közös függőség: Draw2D
 - Rajzolás: IFigure
 - GEF/GMF IFigure elrendezhető Zest-tel
 - De trükközni kell hozzá

GEF3D

GEF3D

- Cél
 - Teljes, 3 dimenziós szerkesztési lehetőségek
 - Létező GEF-es editorok újrahasznosítása
- GEF3D
 - Akadémiai projekt

Architektúra



Problémák

- Dokumentáció nincs
 - GEF nem egyszerű alap
- [@anderiasch](#) (Twitter)
- "Now some Code. If you're not familiar reading Code, you can spend the time figuring out what you're doing on a dev conference." [#gddde](#)