



Tensegrity Software
the interface architect



Tensegrity Visualization Framework

**Modellierung und Visualisierung von Strukturen und
Prozessen**



Überblick

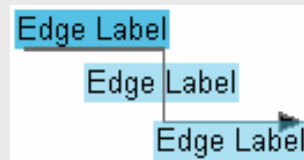
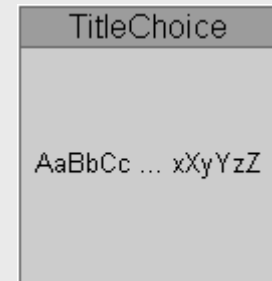
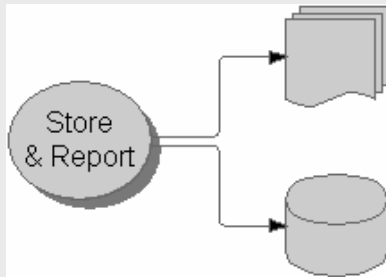
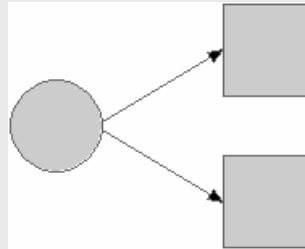
- **Übersicht Features**
 - Visualisierung
 - Modellierung
- **Demo**
- **Architektur und Konzept Graphic Framework**
- **Anwendungsschicht**
- **Eclipse PlugIn**



Visualisierung - Einfache Elemente

- **Kanten**

- Gerade Kante
- Orthogonale Kante
- Kanten mit Text



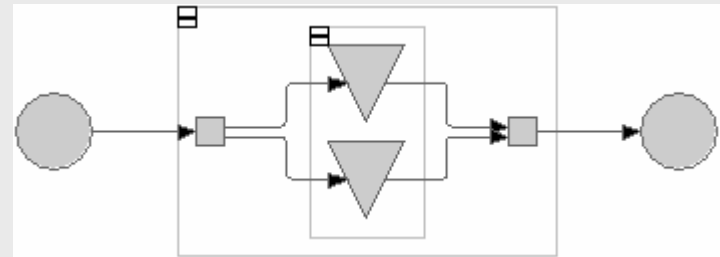
- **Knoten**

- Rechteckige Knoten
- Runde Knoten
- Polygon Knoten
- Knoten mit Text
- Kombinationen



Visualisierung – Komplexe Elemente

- **Knoten**
 - **Gruppierungen**
 - **Tabellen**
 - **Swimlanes**



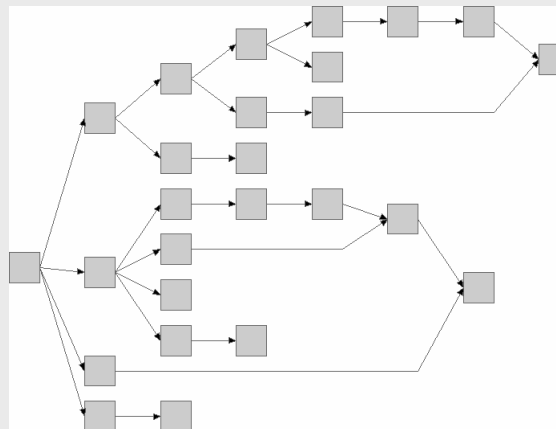
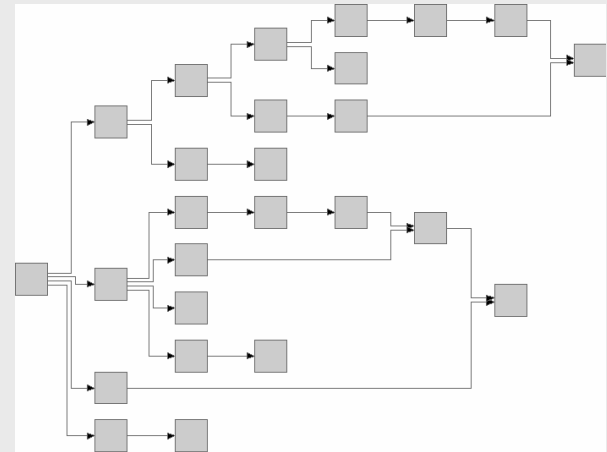
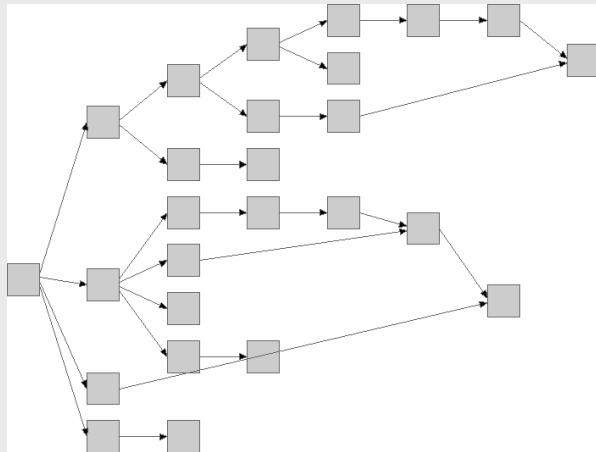
	A	B	C	D	E	F	G
1							
2		cell 1			cell 2		
3							
4		cell 3	cell 4: 2.nd row, third one, go four more!				
5							



Visualisierung – Layout Typen

- **Kanten Layout**

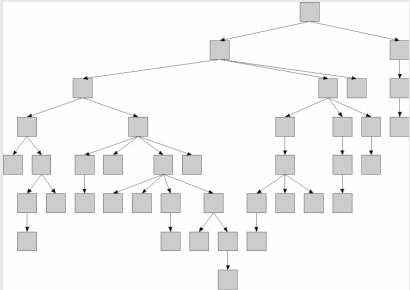
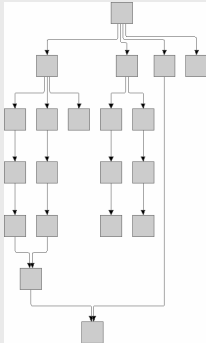
- **Straight**
- **Orthogonal**
- **Kontext sensitiv**



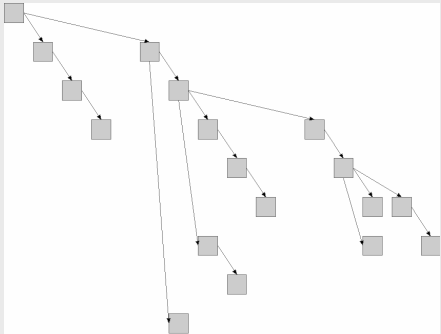


Visualisierung – Layout Typen

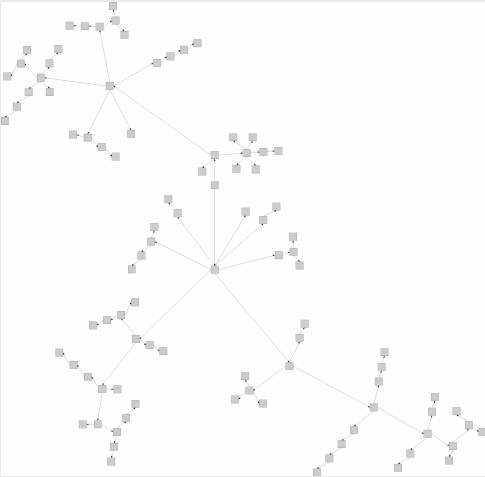
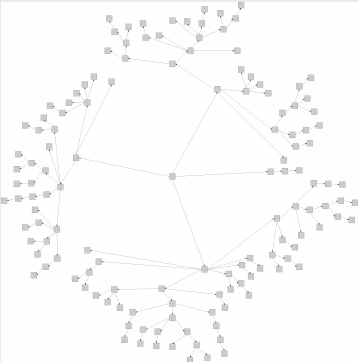
Hierarchisch



Bäume



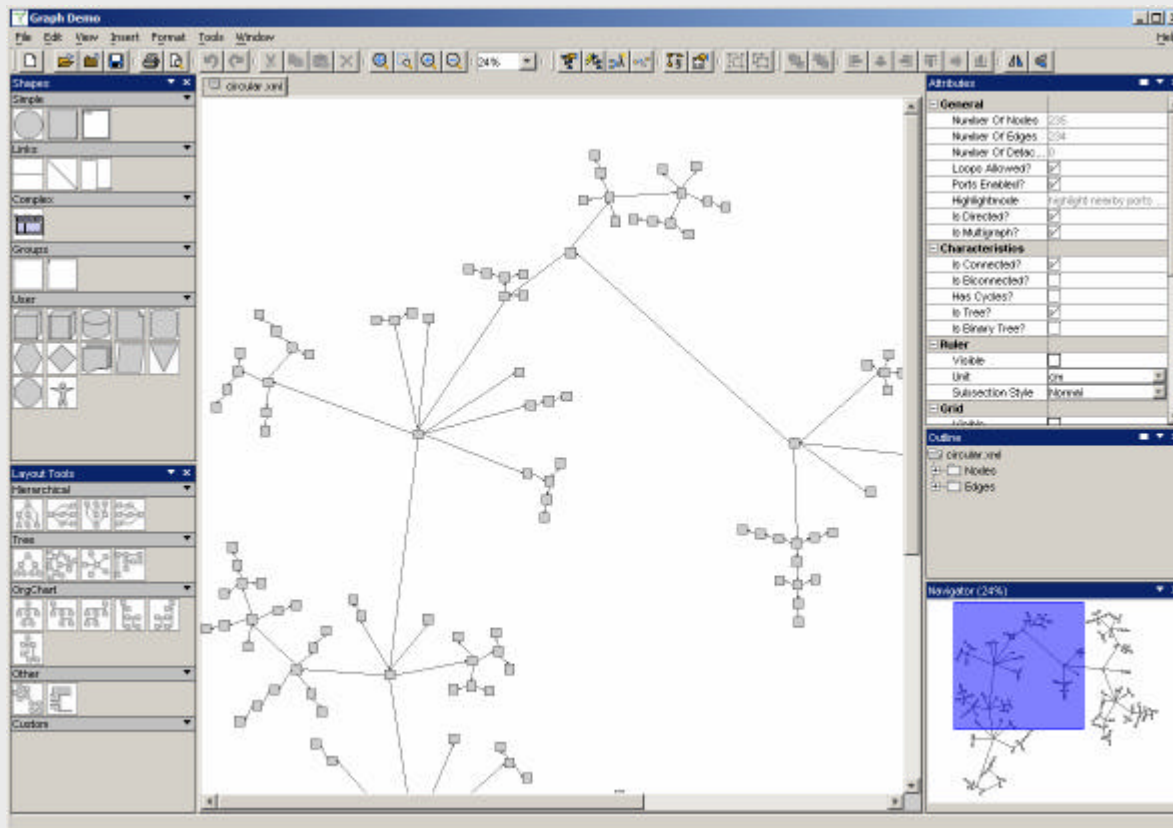
Konzentrisch





Visualisierung Demo

- Graph Demo

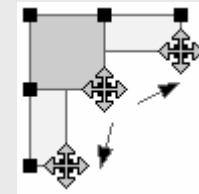




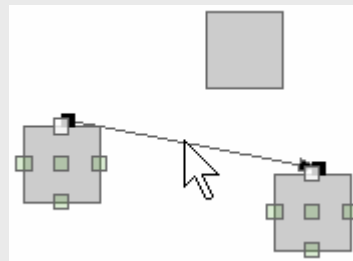
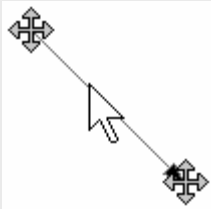
Modellierung – Einfaches modellieren

- Hinzufügen von Elementen aus dem Repository per Drag&Drop
- Hinzufügen von Elementen per Copy / Cut / Paste

- Bewegung und Größenänderung von Knoten



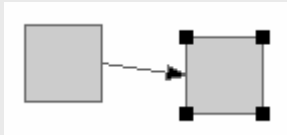
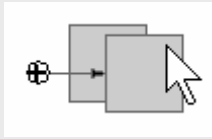
- Bewegung von Kanten



- Anknüpfen von Kanten
 - Highlighting der Anknüpfungspunkte in der Umgebung
 - Highlighting eines berührten Anknüpfungspunktes

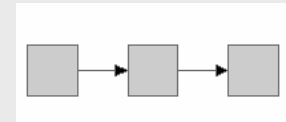
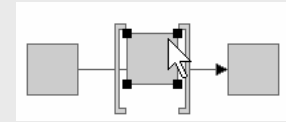
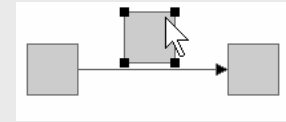


Modellierung – Automatische Erzeugung von Kanten

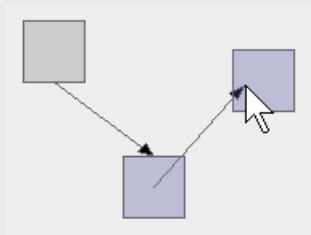
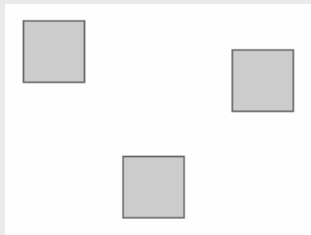


- **Overlay-Edge-Creation**
Wenn ein Knoten über einem anderen Knoten platziert wird

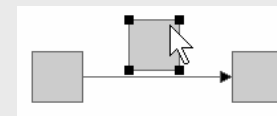
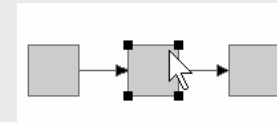
- **Edge-Split**
Wenn ein Knoten auf einer Kante platziert wird



- **Edge-Creation-Mode**
Erzeugung von Kanten per Click&Drag mit der Maus

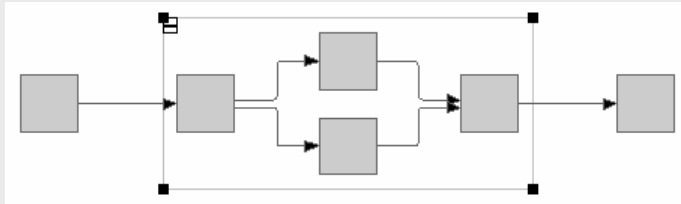


- **Edge-Cascade**
Wenn ein Knoten aus einer transitiven Verbindung gelöst wird

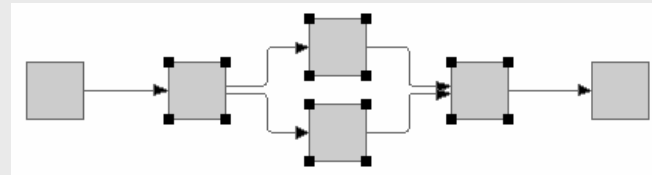




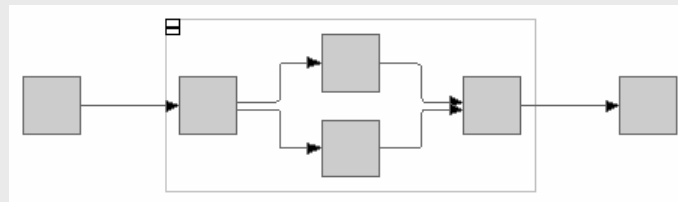
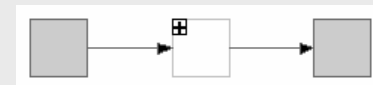
Modellierung – Manipulation von Graphen



- **Gruppierung erzeugen und aufheben**



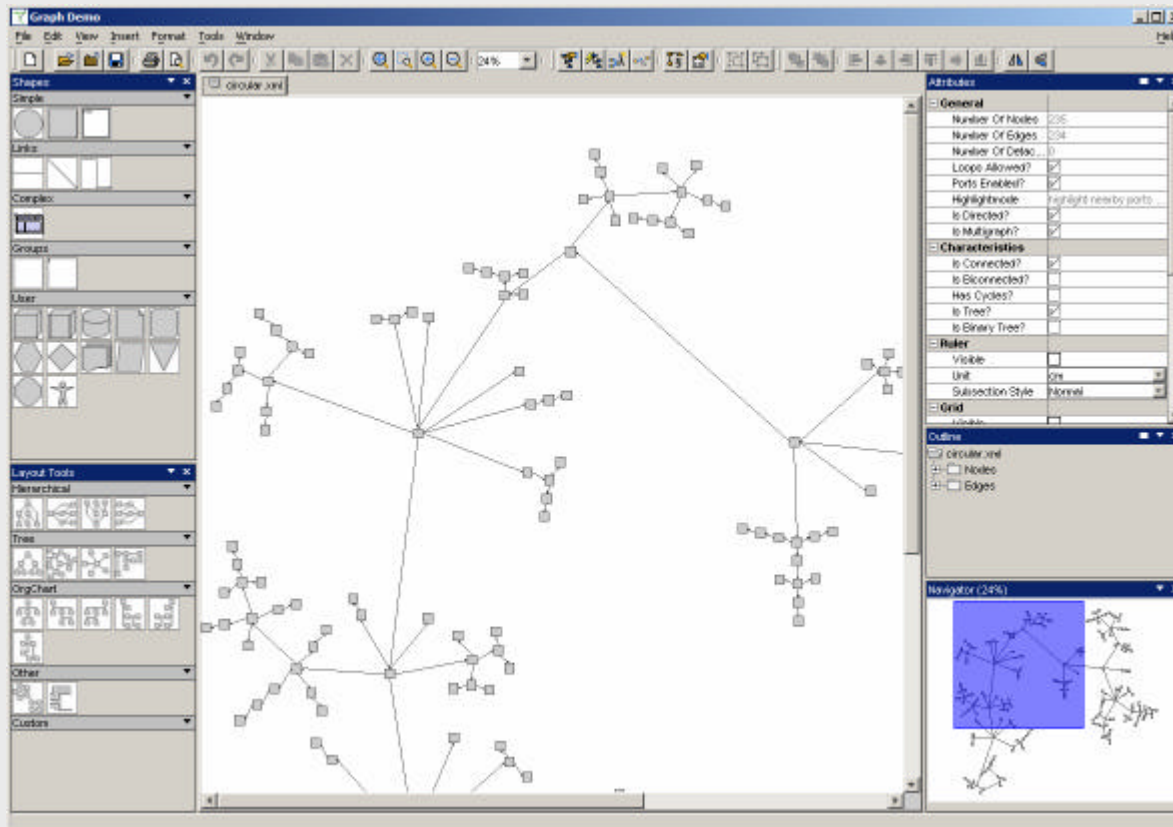
- **Falten eines Graphen**





Modellierung Demo – Graphen erzeugen und bearbeiten

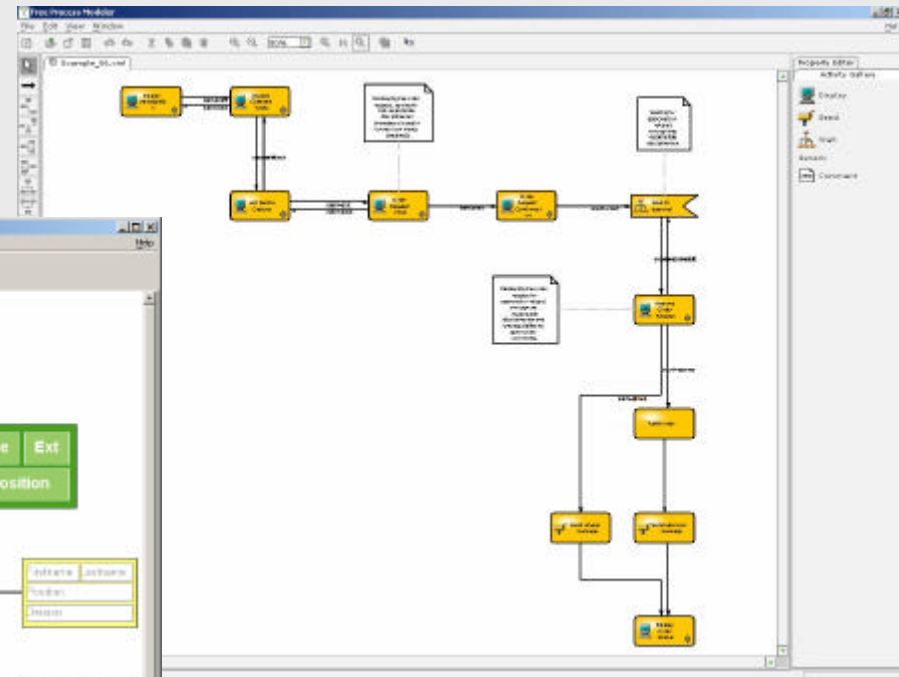
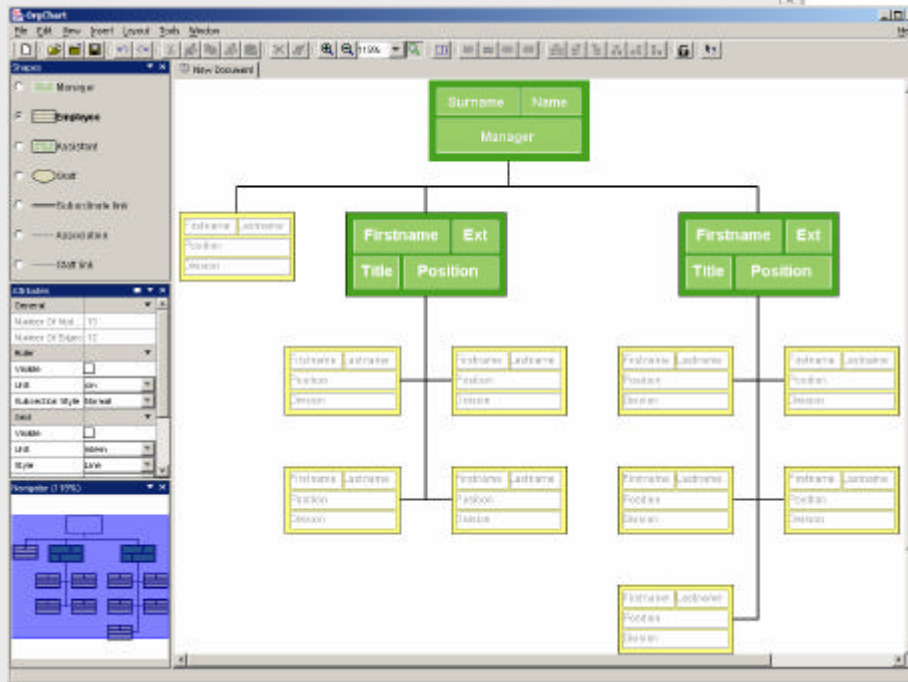
- Graph Demo





Demo - Modellierung von Strukturen

- Free Prozess Modeler



- Organigram

Demo - Modellierung von Prozessen

- **Block Workflow Modeler**

The screenshot displays the 'Basic Workflow Application' interface. The main workspace shows a complex workflow diagram with nested containers and various action blocks. The left sidebar contains a palette of workflow elements, and the bottom panel shows a detailed view of the selected 'Condition' block.

Actortrigger	Eventtrigger	Attribut	Container	Value
Step label			Condition	
Description				
Variable				none...
Comparator				
Value				none...
True branch label				true
False branch label				false



Architektur Graphic Framework – Schichten Modell

Visualisierung

Interaktivität

Modellierung

Zero-,Thin-, Large Clients

Anbindung des Graphic Framework via JSP, Servlet oder Applet

Zero Clients

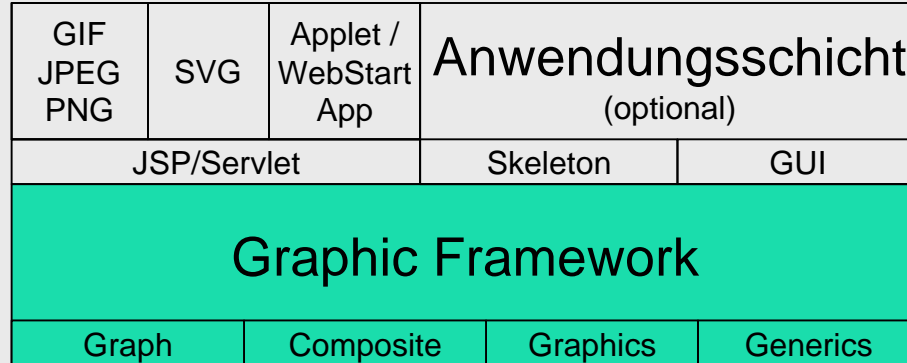
- Exported GIF
- Exported JPEG
- Exported PNG

Thin Client

- Exported SVG

Large Client

- Applet
- WebStart Application



Graphic Framework

- Visualisierung eines Graphen
- Modellierung eines Graphen
- Bereitstellung verschiedener Anordnungstypen
- Bereitstellung verschiedener Algorithmen zur Suche innerhalb eines Graphen (Analyse/Filtern von Graphen)
- Abstraktion vom Window Toolkit
 - AWT/Swing
 - SWT(GEF)/JFace

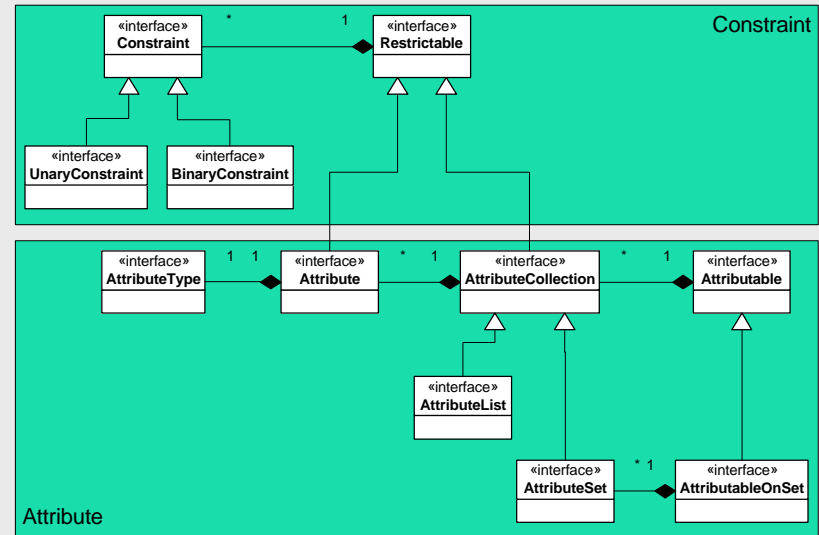
Anwendungsschicht

- Anwendungsrahmen zur schnellen Entwicklung einer Anwendung
- Aufbauend auf Sub API's Skeleton und GUI
- Festlegung des Verhaltens eines Graphen und seiner Elemente
- Definition der Elemente eines Graphen (Repository)
- Optional, da Ansteuerung des Graphic Framework auch direkt möglich



Architektur Graphic Framework - Attribute

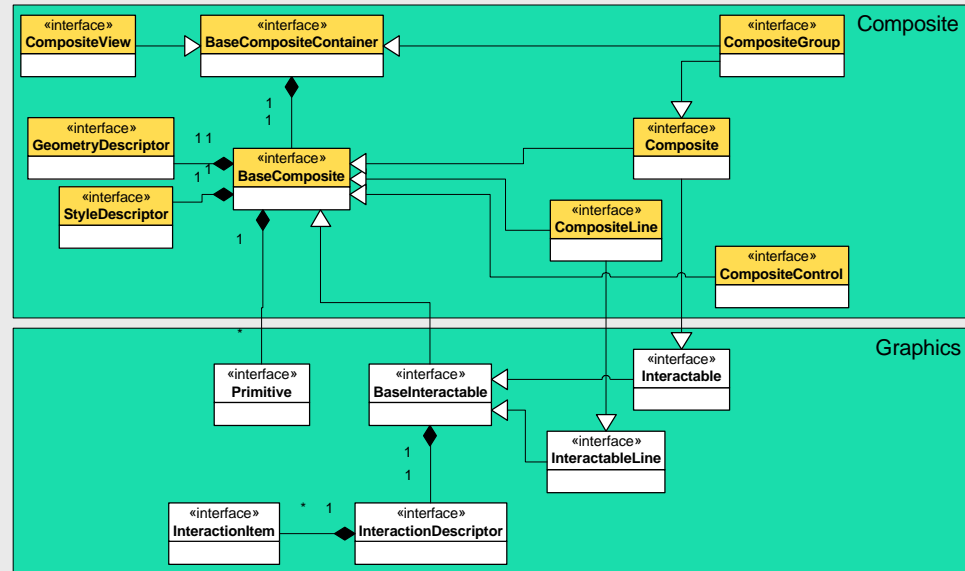
- Name, Value Pair
- Datentypen
- Container (List, Set)
- Verschachtelung
- Restriktion durch Constraints
- Clonen





Architektur Graphic Framework - Composite

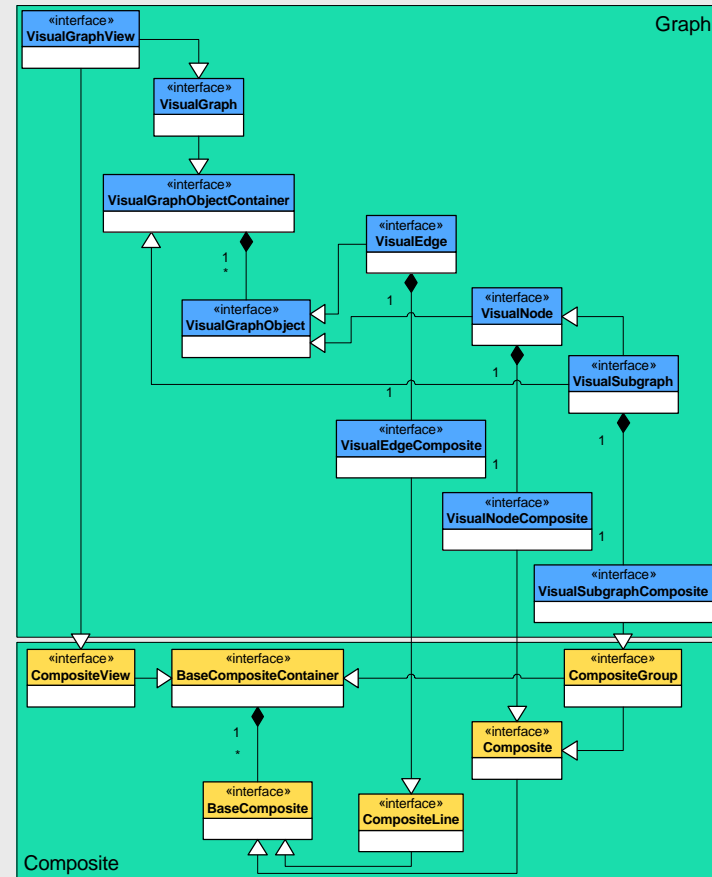
- Unterstützung einfacher graphischer Objekte (Polygone+Text+...)
- Bereitstellung komplexer graphischer Objekte (Tabellen, Swimlanes, ...)
- Unterstützung von Geometrien und Stilen
- Abstraktion vom Window Toolkit



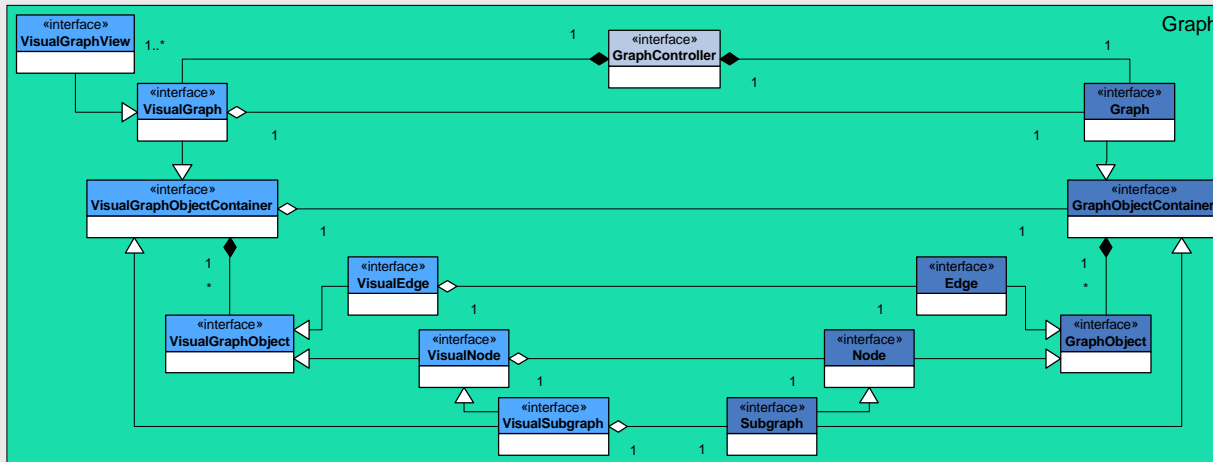


Architektur Graphic Framework - Graph

- Visualisierung durch Composites
- Konfigurierbares Verhalten der Elemente eines Graphen (Rules)
- Zuweisung spezifischer Attribute (Custom Attributes)
- Bereitstellung verschiedener Graph Algorithmen
- Unterstützung von Transaktionen (Undo/Redo)
- Einsprungspunkte für Anwendungen zur Modellierung komplexer Verhalten (Handler)



Architektur Graphic Framework– Graph MVC

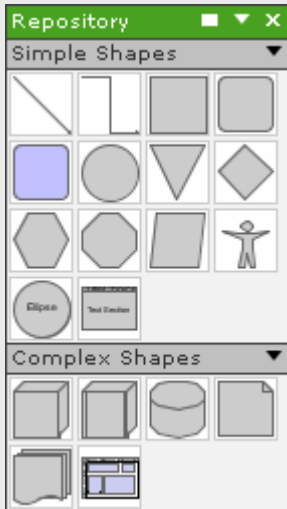
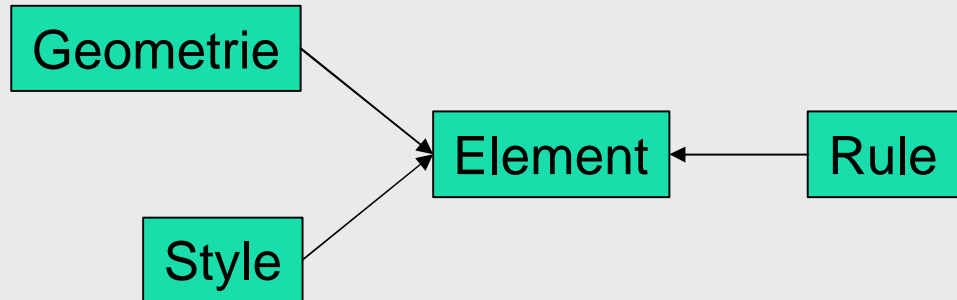


- Unterteilung in Model, View und Controller
- Implementierung für Modell und View
- Verschiedene Controller Typen
 - ClientServerGraphController
 - ModelBasedGraphController



Grundlagen Anwendungsschicht – Repository

Definition eines Elementes:



Aufbau des Repositories:

- Automatisierte Erzeugung der Elemente
- Darstellung der Elemente oder eines Icons
- Unterteilung in Kategorien



Grundlagen Anwendungsschicht – Repository

Definition eines Elementes im Detail

```

<list name="GeometryItem">
  <attribute name="Name" value="DefaultRectangle"/>
  <attribute name="Type" value="Rectangle"/>
</list>
<list name="GeometryDescriptor">
  <attribute name="Name" value="TextEllipse"/>
  <attribute name="Type" value="Composite"/>
  <list name="CoordinateSystem">
    <attribute name="ScaleX" value="0.0, 400.0"/>
    <attribute name="ScaleY" value="0.0, 400.0"/>
  </list>
  <list name="DescriptorItems">
    <list name="DescriptorItem">
      <attribute name="Name" value="Background"/>
      <attribute name="GeometryItemName" value="DefaultRectangle"/>
      <list name="Coordinates">
        <attribute name="Coordinate" value="0, 0"/>
        <attribute name="Coordinate" value="400, 400"/>
      </list>
    </list>
  </list>
  [...]
</list>

```

Geometrie

```

<list name="Rule">
  <attribute name="Name" type="String" value="nrule1"/>
  <attribute name="Type" type="String" value="NodeRule"/>
  <list name="Properties">
    <attribute name="Deletable" type="Boolean" value="true"/>
    <attribute name="Moveable" type="Boolean" value="true"/>
    <attribute name="Selectable" type="Boolean" value="true"/>
    <attribute name="Resizable" type="Boolean" value="true"/>
    <attribute name="IndegreeMinimum" type="Integer" value="0"/>
    <attribute name="IndegreeMaximum" type="Integer" value="20"/>
    <attribute name="OutdegreeMinimum" type="Integer" value="0"/>
    <attribute name="OutdegreeMaximum" type="Integer" value="20"/>
    <attribute name="InstanceMinimum" type="Integer" value="0"/>
    <attribute name="InstanceMaximum" type="Integer" value="20"/>
    <attribute name="CascadeTo1NodeDelete" type="Boolean" value="true"/>
    <attribute name="CascadeDeleteEdges" type="Boolean" value="false"/>
    <attribute name="DrawPorts" type="Boolean" value="false"/>
    <attribute name="DetachOnMove" type="Boolean" value="false"/>
    <attribute name="AcceptLooseNodesIfSubgraph" type="Boolean" value="true"/>
    <attribute name="DropHandlerName" type="String" value=""/>
    <attribute name="SubgraphRemoveHandlerName" type="String" value=""/>
  </list>
</list>

```

```

<list name="StyleItem">
  <attribute name="Name" value="GrayRectangle"/>
  <attribute name="Type" value="Rectangle"/>
  <set name="Attributes">
    <set name="Fill">
      <attribute name="background" type="Color" value="204, 204, 204"/>
    </set>
    <set name="Border">
      <attribute name="color" type="Color" value="102, 102, 102"/>
    </set>
  </set>
</list>
<list name="StyleDescriptor">
  <attribute name="Name" value="NiceNode"/>
  <attribute name="Type" value="Composite"/>
  <list name="StyleItems">
    <attribute name="StyleItem" value="GrayRectangle"/>
  </list>
  [...]
</list>

```

Stil

Verhalten

```

<list name="NodeElement">
  <attribute name="Name" type="String" value="EmptySubgraph"/>
  <attribute name="Geometry" type="String" value="EmptySubgraph"/>
  <attribute name="Style" type="String" value="TenseEmptySubgraph"/>
  <attribute name="Rule" type="String" value="EmptySubgraphRule"/>
</list>

```

Element



Grundlagen Anwendungsschicht – Attribute Tree

- Anzeige der Attribute eines oder mehrerer Elemente
- Modifikation der Attribute direkt im AttributeTree oder innerhalb von Dialogen
- Kategorisierung der Attribute in mehreren Levels
- Unterstützung von Custom Editoren/Dialogen
- Umbenennung der Attribute
- Ausblenden der Attribute

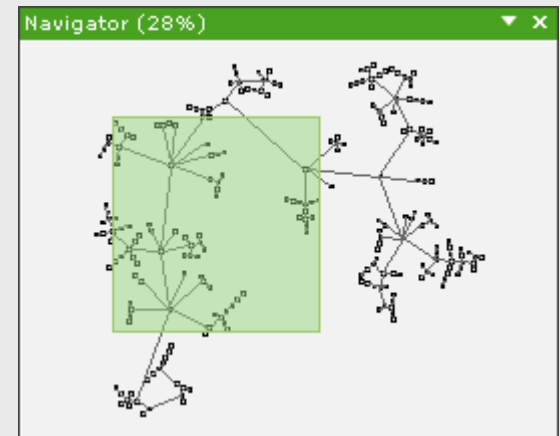
The screenshot shows a window titled 'Attributes' with a tree view of attributes for a 'CylinderNode'. The tree is expanded to show the 'CylinderPolygon' node, which is highlighted in green. The 'CylinderPolygon' node has several sub-attributes, including 'Fill', 'Style', 'Gradient', 'Pattern', 'BackColor', 'ForeColor', 'Transparent', and 'Transparency'. The 'Fill' attribute is currently set to a light gray color.

Attributes	
[-] CylinderNode	
[-] ModelNode	
Description	<empty>
[-] ViewNode	
Rule	nrule1
[-] Composite	
[-] Geometry	
X	3070
Y	3293
Width	400
Height	400
MinimumWidth	1
MinimumHeight	1
[-] Appearance	
Shadow	<input type="checkbox"/>
[-] TopPolygon	
+ Border	<input type="text"/> ...
+ Fill	<input type="text"/> ...
[-] CylinderPolygon	
[-] Border	<input type="text"/> ...
Style	Continuous
Color	<input type="text"/> ...
Weight	1
[-] Fill	<input type="text"/> ...
Style	Solid
Gradient	Vertical
Pattern	Solid
BackColor	<input type="text"/> ...
ForeColor	<input type="text"/> ...
Transparent	<input type="checkbox"/>
Transparency	50



Grundlagen Anwendungsschicht – Navigator

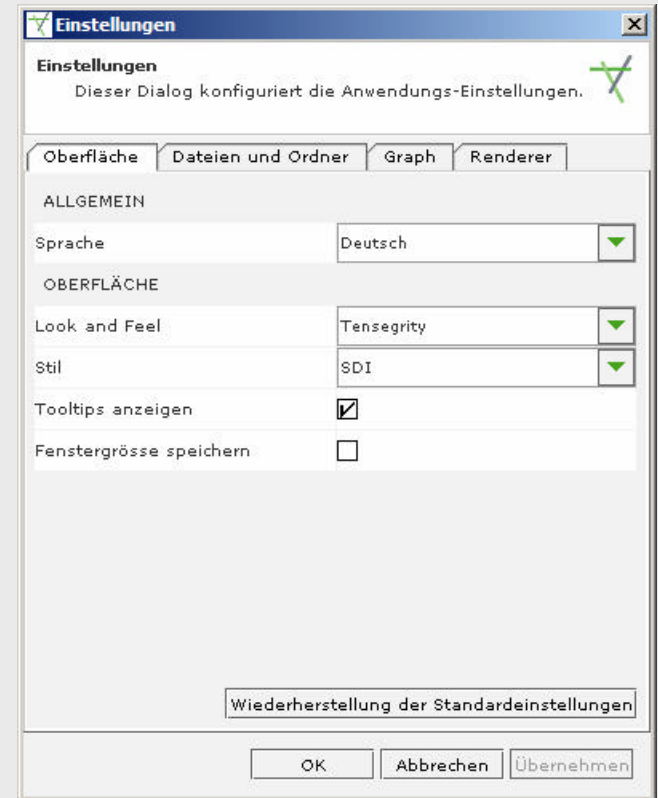
- Übersicht des gesamten Graphen
- Anzeige des aktuell sichtbaren Bereiches des Graphen
- Verschieben des aktuell sichtbaren Bereiches des Graphen
- Vergrössern/ Verkleinern des aktuell sichtbaren Bereiches des Graphen (ändern der Skalierung)
- Konfiguration des ‘Level of Details‘ der angezeigten Elemente





Grundlagen Anwendungsschicht – Preferences

- **Persistente Voreinstellungen zur Konfiguration einer Anwendung**
- **Sprache**
- **LookAndFeel**
- **Nützliche Einstellungen für Dateien und Ordern**
- **Globale Einstellungen der Graphen**
- **‘LevelOfDetail‘ für View und Navigator**





Grundlagen Anwendungsschicht – Skeleton

Toolbar

Repository

Layout Tools

Custom Container

The screenshot shows the GraphPad software interface. The main window displays a complex graph with a central node and several branches. The interface includes a menu bar (File, Edit, View, Insert, Format, Tools, Window, Help), a toolbar with various icons, and several docked panels on the left and right. The left panels include a Repository (Simple Shapes and Complex Shapes), Layout Tools (Org Chart, Circular, Energy), and an Outline (listing labels like label (864), label (870), etc.). The right panels include an AttributeTree (General, Characteristics, Rules, Grid, Geometry) and a Navigator (28%) showing a smaller view of the graph.

Multiple Dokumente

Graph View

AttributeTree

Navigator



Eclipse PlugIn

Toolbar

Multiple Dokumente

Standard Eclipse Container

Graph View

Repository

AttributeTree

Navigator

