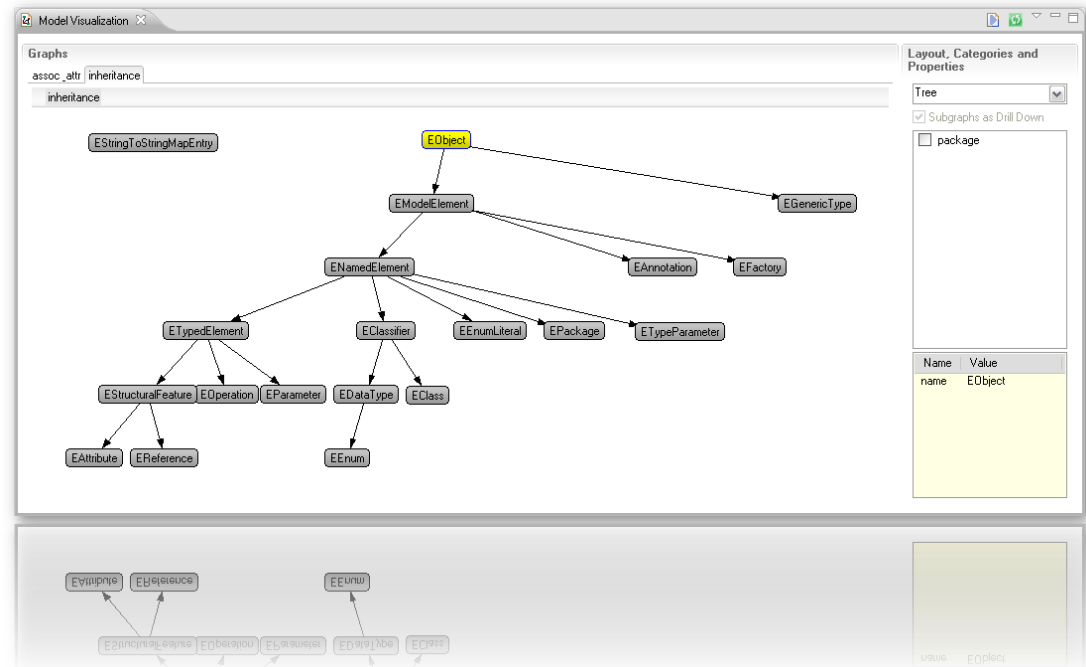


# Modellvisualisierung leicht gemacht

JUG Karlsruhe  
Lightning Talks  
24.03.2010

Daniel Weber



# emfmodelvisualizer



**MDSD**



**Eclipse**

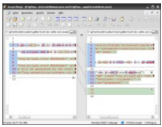


**Grafische Ansicht gewünscht**

# Warum nicht grafisch modellieren?

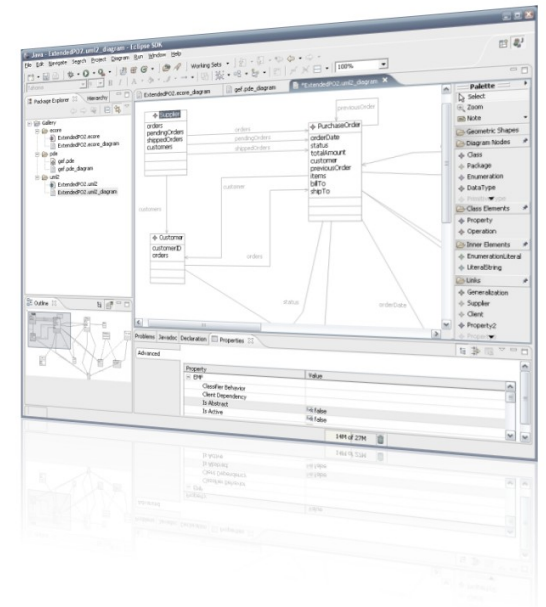


# Entwicklungsaufwand



# Diff/Merge

● ● ●



# Textuell modellieren & visualisieren

```
package org.xtext.example;

import org.eclipse.emf.ecore.EPackage;
import org.eclipse.emf.ecore.resource.Resource;

/**
 * Generated from StandaloneSetup.xpt!
 */

public class MyDslStandaloneSetup implements ISetup {

    public Injector createInjectorAndDoEMFRegistration() {
        TerminalsStandaloneSetup.doSetup();
        Injector injector = createInjector();
        register(injector);
        return injector;
    }

    public Injector createInjector() {
        return Guice.createInjector(new MyDslRuntimeModule());
    }

    public void register(Injector injector) {
        if(EPackage.Registry.INSTANCE.containsKey(
            "http://www.xtext.org/example/MyDsl")) {
            EPackage.Registry.INSTANCE.put(
                "http://www.xtext.org/example/MyDsl",
                MyDslPackage.eINSTANCE);
        }
    }
}
```

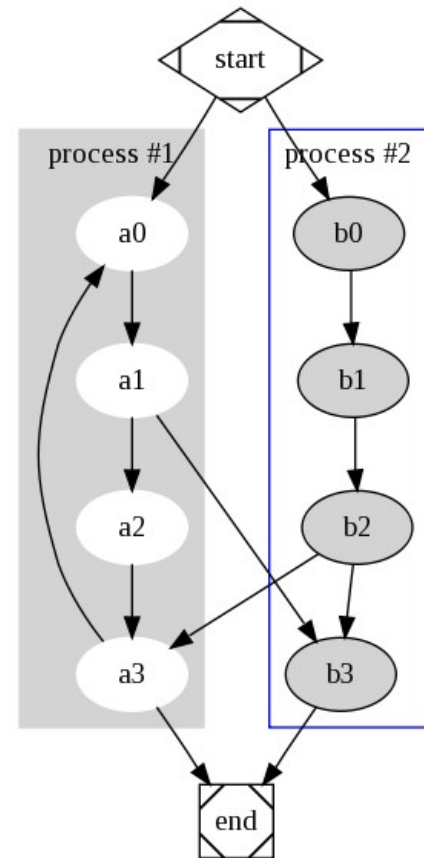
&



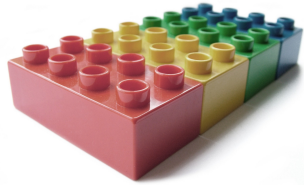
# Graphviz.org

```
digraph G {  
  
  subgraph cluster_0 {  
    style=filled;  
    color=lightgrey;  
    node [style=filled,color=white];  
    a0 -> a1 -> a2 -> a3;  
    label = "process #1";  
  }  
  
  subgraph cluster_1 {  
    node [style=filled];  
    b0 -> b1 -> b2 -> b3;  
    label = "process #2";  
    color=blue  
  }  
  
  start -> a0;  
  start -> b0;  
  a1 -> b3;  
  b2 -> a3;  
  a3 -> a0;  
  a3 -> end;  
  b3 -> end;  
  
  start [shape=Mdiamond];  
  end [shape=Msquare];  
}
```

dot



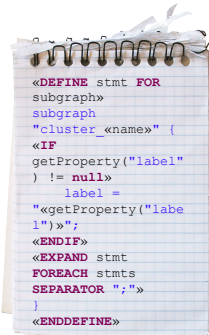
# Dot Generator



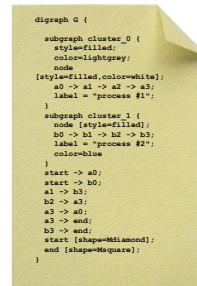
Modell



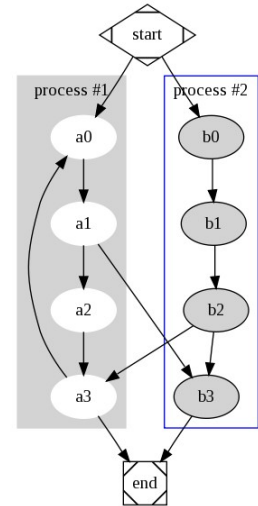
Generator



Template

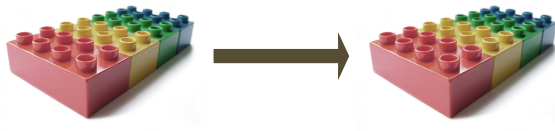
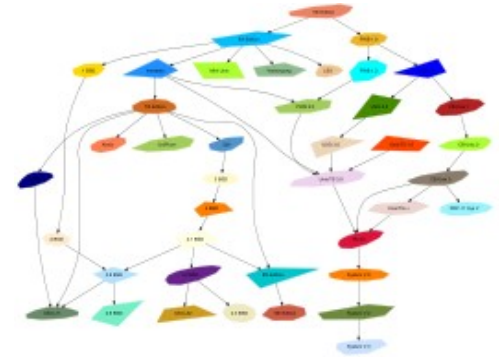


.dot  
Datei

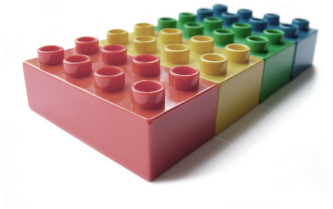


# Dot Metamodell

- **Ursprung in oaw**
- **Xtext-basiert**
- **Dot-spezifisch**
- **Ermöglicht model2model**



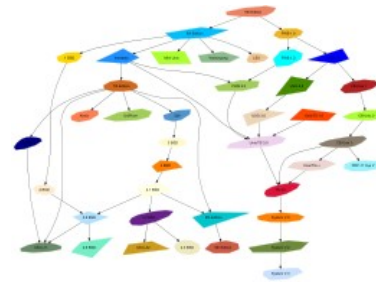
# Dot Modelltransformation



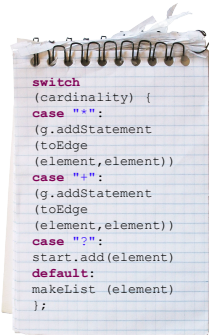
Modell



Transformator



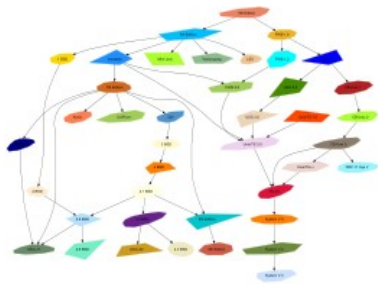
Dot Modell



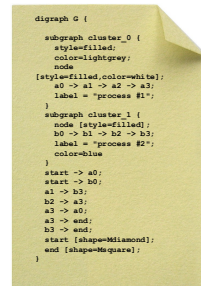
Modell-  
transformation



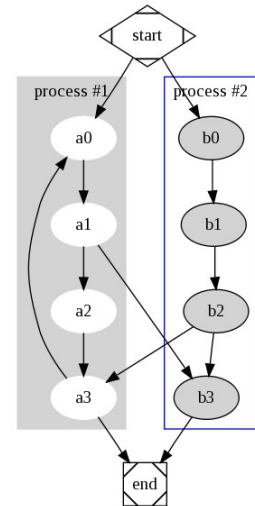
# emfmodelvisualizer, übernehmen Sie



Dot Modell

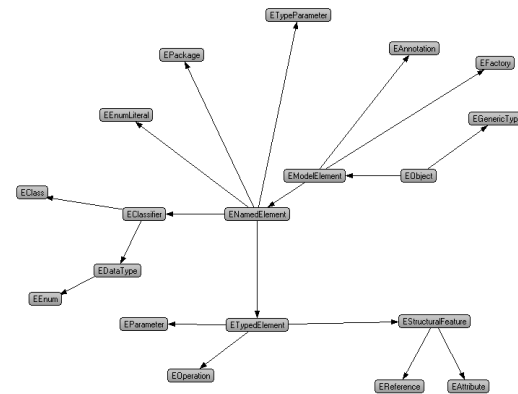


.dot  
Datei

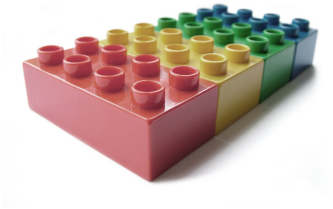


# Graph Metamodell

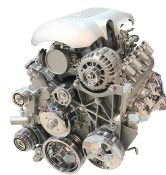
- **Graphen, Knoten, Kanten**
- **Xtext-basiert**
- **Generisch**



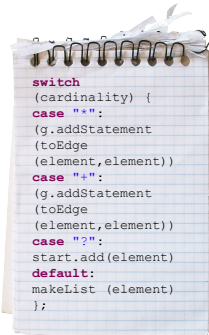
# Model to graph



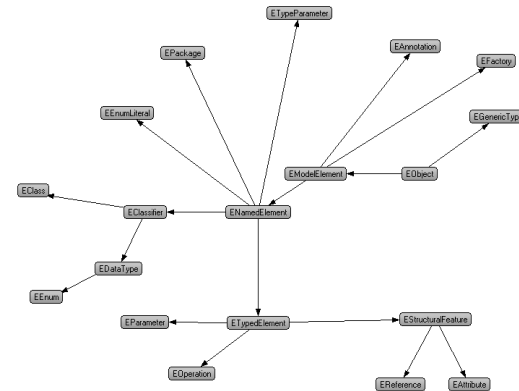
Modell



Transformator

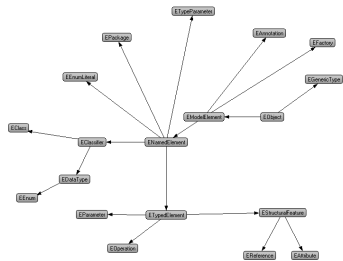


Modell-  
transformation



Graph Modell

# Graph Model „Backends“



## Graph Modell



## Validierung



## zest viewer

# xmind

# ubigraph

• • •

(dot)

# Zest Viewer

## Graphs

assoc\_attr inheritance

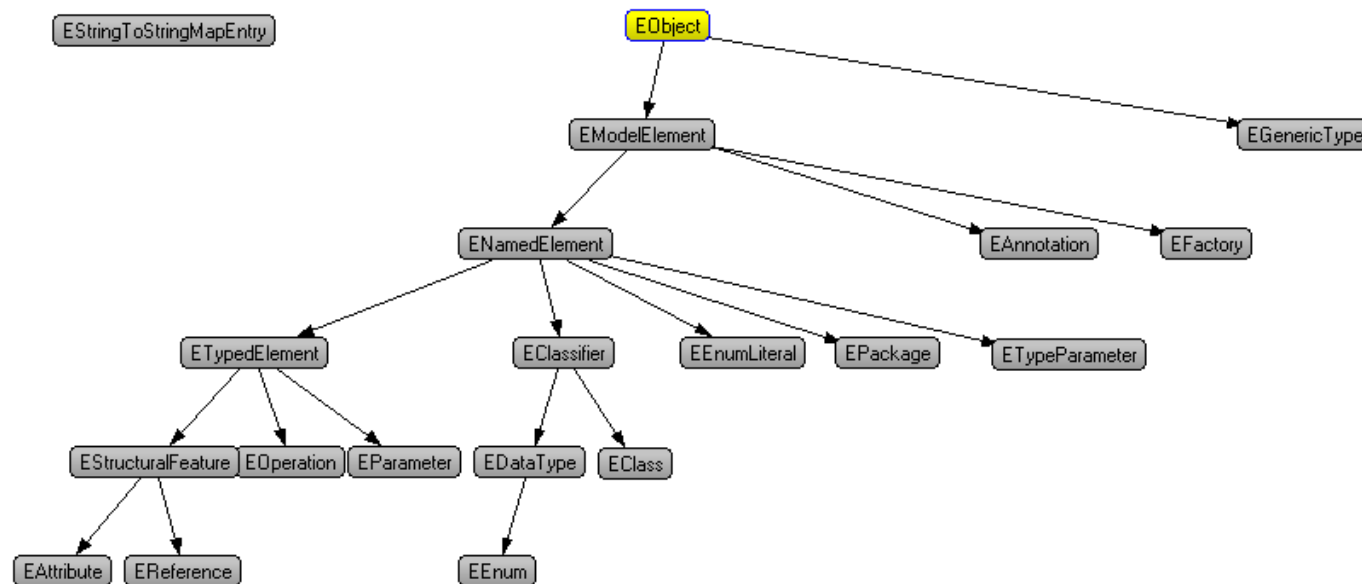
inheritance

## Layout, Categories and Properties

Tree

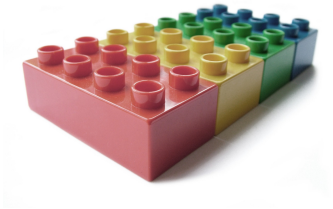
☒ Subgraphs as Drill Down

☐ package



Name	Value
name	EObject

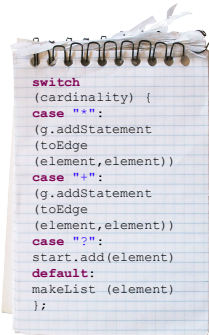
# Blockdiagramme



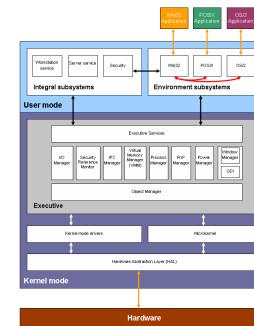
## Modell



# Transformer

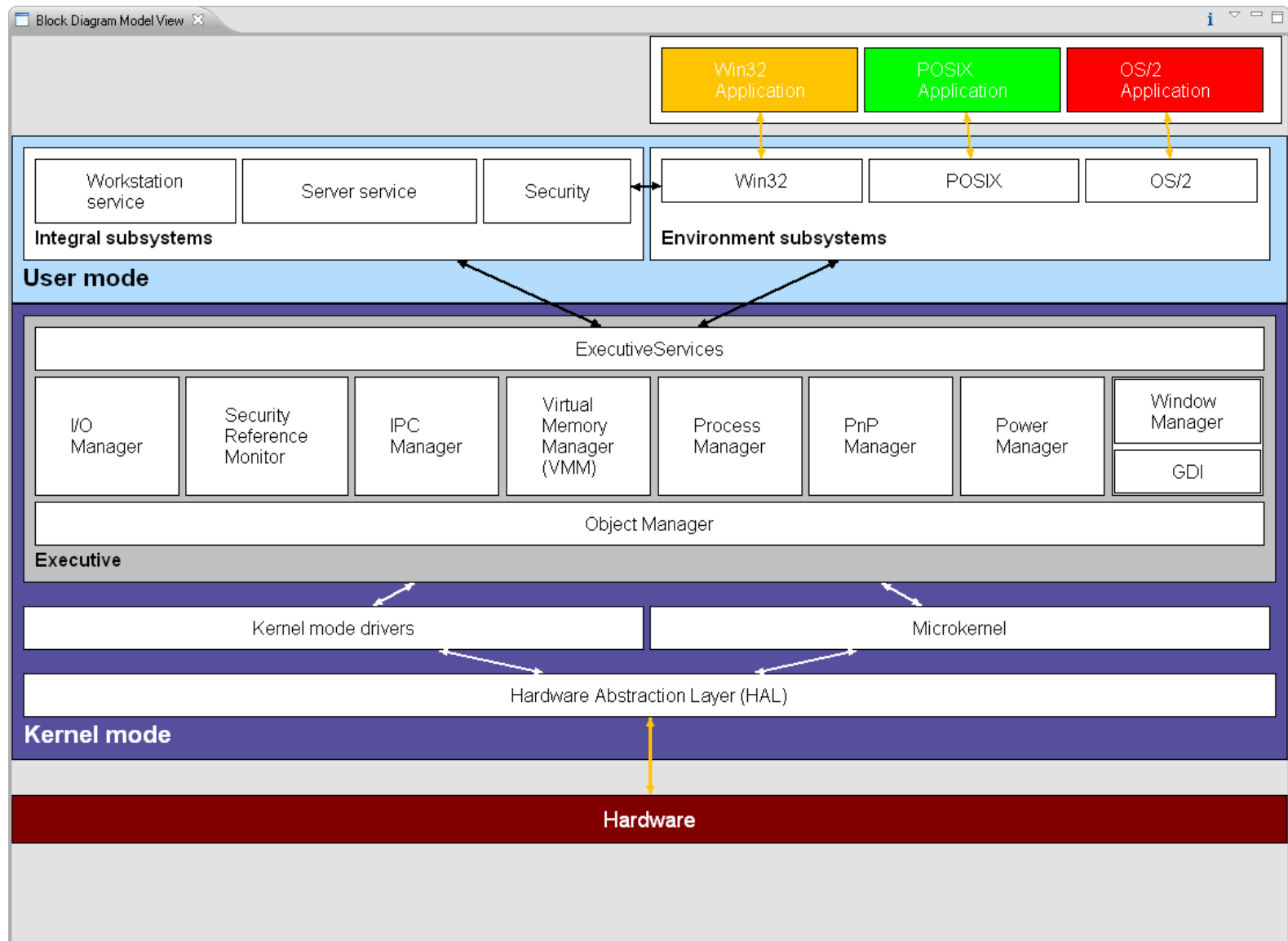


## Modell- transformation



## Blockdiagram Modell

# Blockdiagramm View



# Ausblick

- **Eclipse helios (3.6)**
- **graphmm2dot**
- **Weitere graphmm backends?**
- **Ideen/Anregungen/Bug reports...**



# Links

- [emfmodelvisualizer.googlecode.com](https://emfmodelvisualizer.googlecode.com)
- [graphviz.org](https://graphviz.org)
- [eclipse.org/gef/zest](https://eclipse.org/gef/zest)
- [eclipse.org/Xtext](https://eclipse.org/Xtext)
- [danielweber.github.com](https://danielweber.github.com)

# Diskussion

