

# Postfuse Eclipse Plugin zum Visualisieren von Graphen

Bachelorpraktikum - Review 2 - Design

05.02.2007

Gruppe: G<sup>222</sup>

# Gliederung

Design Überlegungen

Architektur

Use-Cases Realisierungen

Stand der Dinge

# Gliederung

Design Überlegungen

Architektur

Use-Cases Realisierungen

Stand der Dinge

# Design Überlegungen

## Benutzercharakteristik - GUI

### Der GUI-Benutzer

- ▶ weiß was ein Graph ist
- ▶ ist in der Lage Eclipse zu benutzen
- ▶ ist in der Lage ein Plugin von einer Update Site zu installieren
- ▶ ist in der Lage eine GUI zu benutzen

# Design Überlegungen

## Benutzercharakteristik - API

### Der API-Benutzer

- ▶ ... (wie GUI-Benutzer)
- ▶ kann Java Programmieren
- ▶ ist in der Lage ein Eclipse Plugin zu erstellen
- ▶ weiss wie man auf die APIs von anderen Plugins zugreift

# Design Überlegungen

## Standards

- ▶ Betrifft hauptsächlich Dateiformate
- ▶ GraphML für Graph-Dateien → XML konform erweitern
- ▶ Bild Export: SVG, PNG
- ▶ Plugin-Standards
- ▶ Update Site Format

# Design Überlegungen

## sonstige Anforderungen

- ▶ QS: Funktionalität
- ▶ QS: Benutzbarkeit
- ▶ freie Lizenz (BSD)
- ▶ ...

# Gliederung

Design Überlegungen

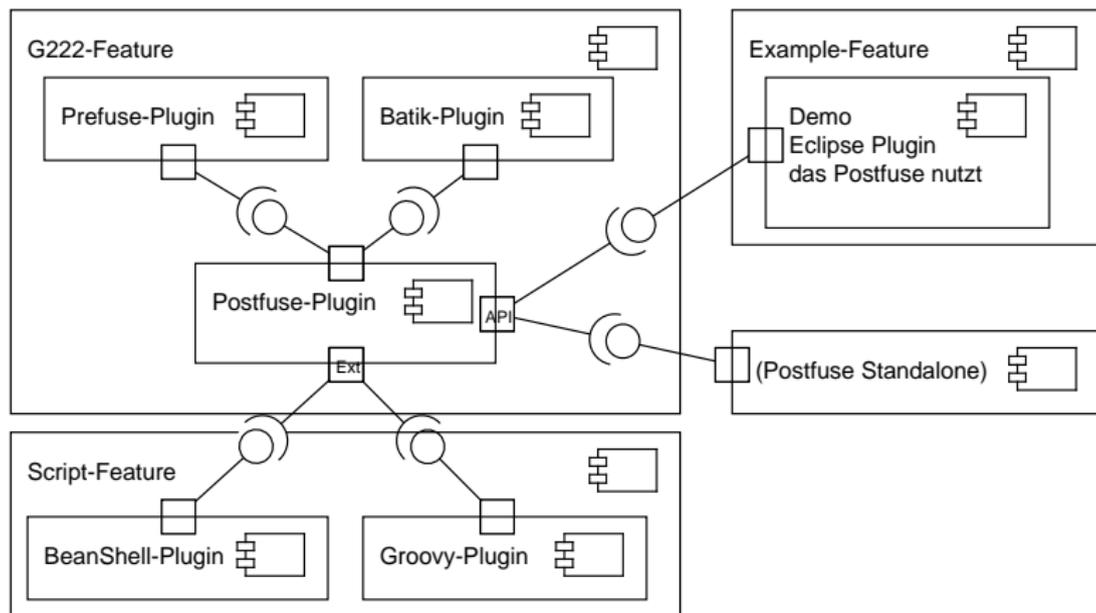
**Architektur**

Use-Cases Realisierungen

Stand der Dinge

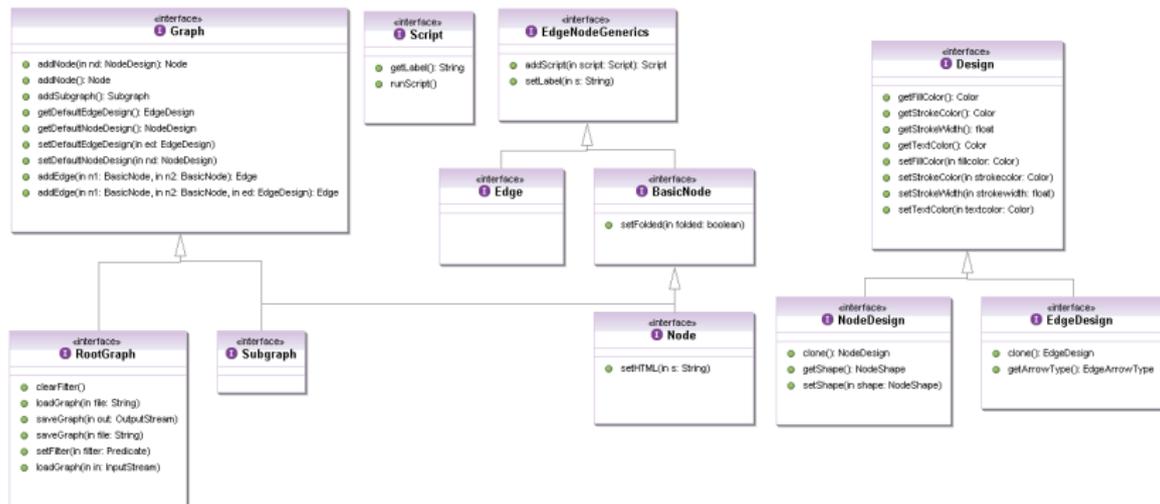
# Architektur

## Subsysteme und ihre Aufgaben



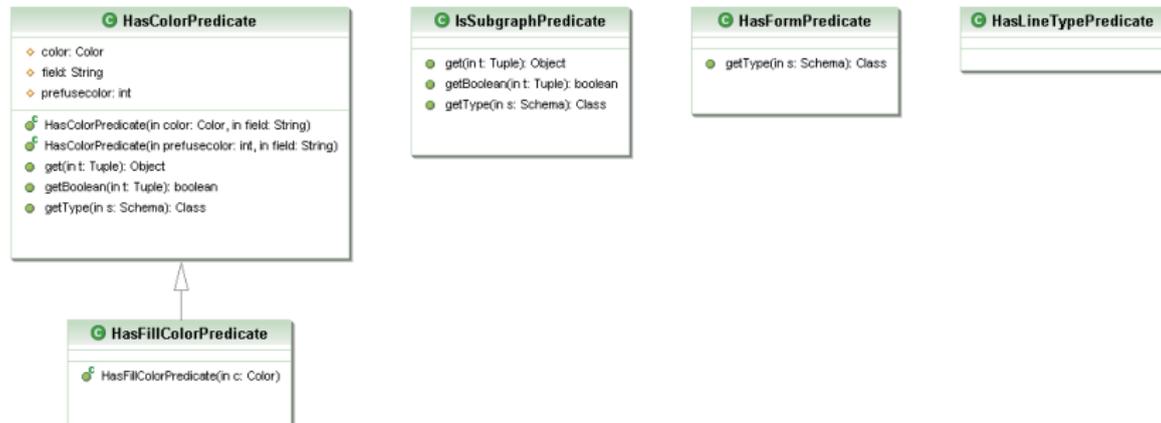
# Architektur

## Die API



# Architektur

## Die API - Filter



# Architektur

## Die API des Plugins

### G222Plugin

  PLUGIN\_ID: String

  plugin: G222Plugin

  G222Plugin()

  createGraph(): RootGraph

  getDefault(): G222Plugin

  getImageDescriptor(in path: String): ImageDescriptor

  saveGraph(in g: RootGraph, in containerName: String, in fileName: String)

  showGraphInEditor(in g: RootGraph)

  showGraphInView(in g: RootGraph)

  saveGraph(in g: RootGraph, in file: IFile): boolean

  saveGraph(in g: RootGraph, in file: IFile, in monitor: IProgressMonitor)

 stop(in context: BundleContext)

 start(in context: BundleContext)

# Architektur

## Die GUI

The screenshot shows the Eclipse IDE interface for a project named "G222 Multi-page Editor". The main editor area is split into two panes. The left pane displays a diagram with nodes labeled #0 through #7 and text elements like "daselbe erste String" and "dies ist mehrzeiliger Text". The right pane shows a content editor with a diagram of nodes #0, #1, and #2, and a text box containing "Content of #1". A tooltip over the text box reads "Eintrag 1", "this is a node", and "Gerätet". The bottom status bar shows "Auswerten" and "britannien: Polizei". The system tray at the bottom right indicates the date and time: "22.03.18 31.01.2007".

Java - G222 Multi-page Editor - Eclipse SDK

File Edit Navigate Search Project Run Editor Menu Window Help

Package... Hierarchy G222 Multi-page Edi... G222 Multi-page Edi... G222 Multi-page Editor x Out... Gra...

xml ts  
test.gml

View/Properties

Problems Javadoc Declaration  
0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location
-------------	----------	------	----------

Auswerten britannien: Polizei 22.03.18 31.01.2007

- ▶ Andere Skriptsprachen
- ▶ Eigene Filter (Prädikate)
- ▶ Umgebungen in denen die Skripts ausgeführt werden
- ▶ Prefuse Erweiterbarkeit (teilweise) beibehalten
  - ▶ Layouter
  - ▶ Design der Elemente

# Gliederung

Design Überlegungen

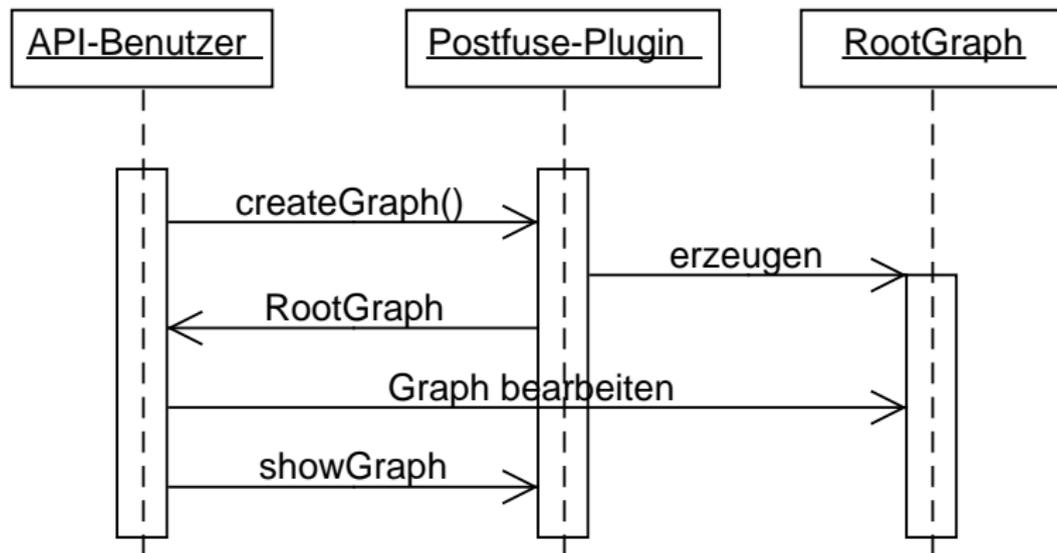
Architektur

**Use-Cases Realisierungen**

Stand der Dinge

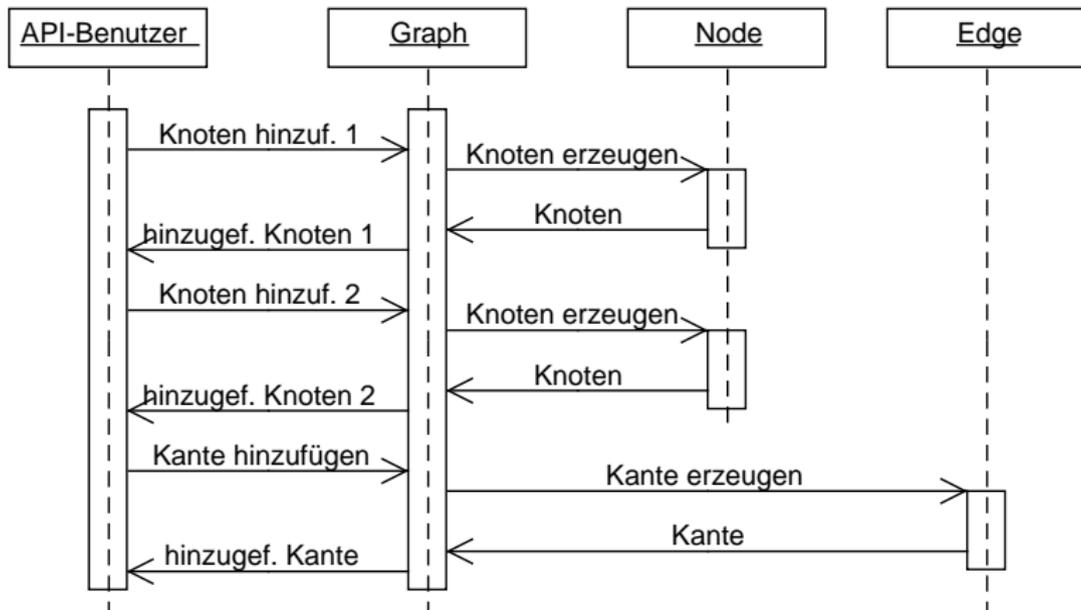
# Use-Cases Realisierungen

Graph erstellen und anzeigen



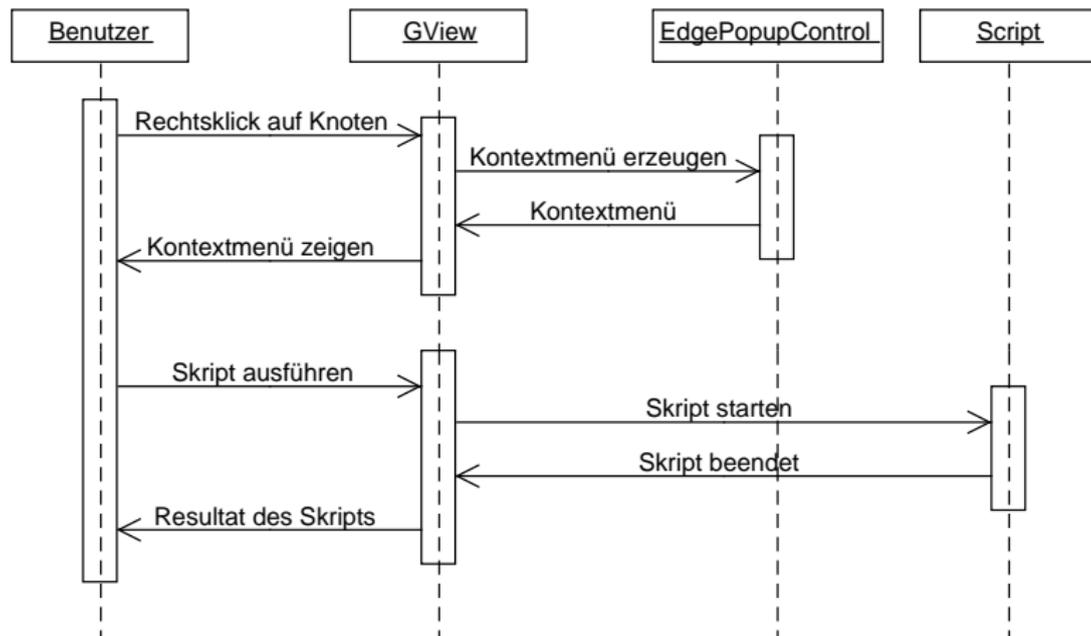
# Use-Cases Realisierungen

## Knoten & Kanten erstellen



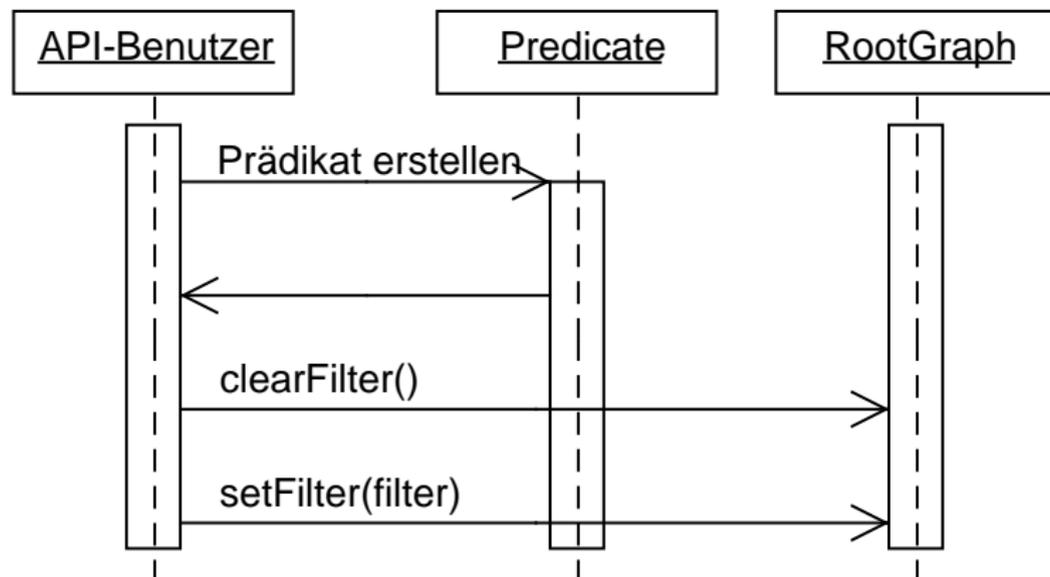
# Use-Cases Realisierungen

## Skripte



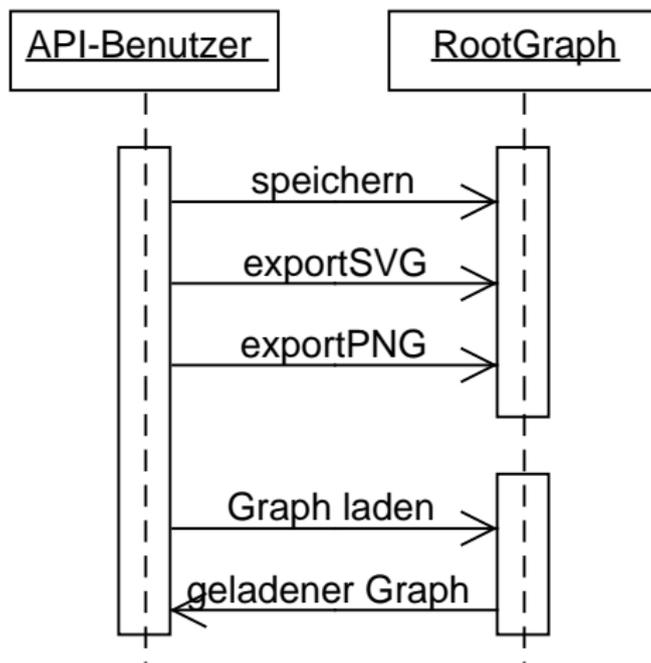
# Use-Cases Realisierungen

## Filter



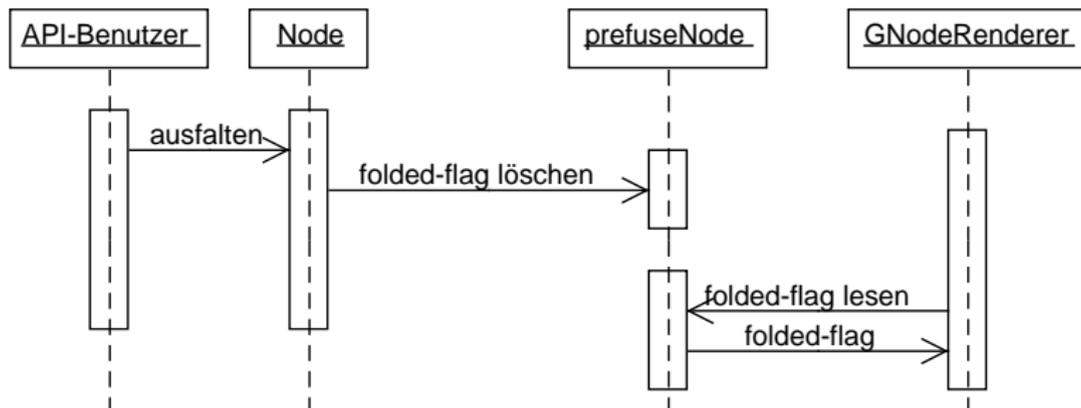
# Use-Cases Realisierungen

Laden / Speichern



# Use-Cases Realisierungen

## Knotenfaltung



# Gliederung

Design Überlegungen

Architektur

Use-Cases Realisierungen

**Stand der Dinge**

# Stand der Dinge

Was geht bereits / Demo

- ▶ Export: SVG, PNG
- ▶ Laden / Speichern GraphML (noch nicht endgültiges Dateiformat)
- ▶ Anzeigen von Graphen ohne Subgraphen incl. Zoom und Pan
- ▶ Auf- und Zuklappen von Knoten
- ▶ Skripte ausführen
- ▶ Eclipse: Update Site Installation
- ▶ Eclipse: MultiView mit \*.gml Dateiassoziation
- ▶ ...

# Stand der Dinge

Was geht noch nicht?

- ▶ Subgraphen (Layout)
- ▶ HTML in Nodes
- ▶ MacOSX
- ▶ ...

# Stand der Dinge

## Arbeitszeiten

- ▶ 1000 h Zielmarke
- ▶ 390 h Bisher
  - ▶ 5% Qualitätssicherung
  - ▶ 20% Treffen für Designfragen & Pariprogramming
  - ▶ 25% Ausarbeitung des Designs
  - ▶ 20% Toolsuche und Installation
  - ▶ 30% Implementierung
- ▶ Schwerpunkt auf die Semesterferien gelegt

# Stand der Dinge

## Weitere Planungen

- ▶ Subgraphen Komponenten erfordern große Änderungen (neue Module)
  - ▶ Neuer Layouter (Mehrfachkanten, nichtüberlappende Knoten, Subgraphen)
  - ▶ Renderer für Subgraphen
- ▶ bessere Eclipse Integration (z.B. Extension-Points)

# Fragen?

