

Postfuse

Eclipse Plugin zum Visualisieren von Graphen

Bachelorpraktikum - Review 2 - Design

05.02.2007

Gruppe: G²²²

Gliederung

- 1 Design Überlegungen
- 2 Architektur
- 3 Use-Cases Realisierungen
- 4 Stand der Dinge

Gliederung

- 1 Design Überlegungen
- 2 Architektur
- 3 Use-Cases Realisierungen
- 4 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 - 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 - 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 - 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

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

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

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

Standards

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

Design Überlegungen

Standards

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

Design Überlegungen

Standards

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

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)
- ...

Design Überlegungen

sonstige Anforderungen

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

Design Überlegungen

sonstige Anforderungen

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

Design Überlegungen

sonstige Anforderungen

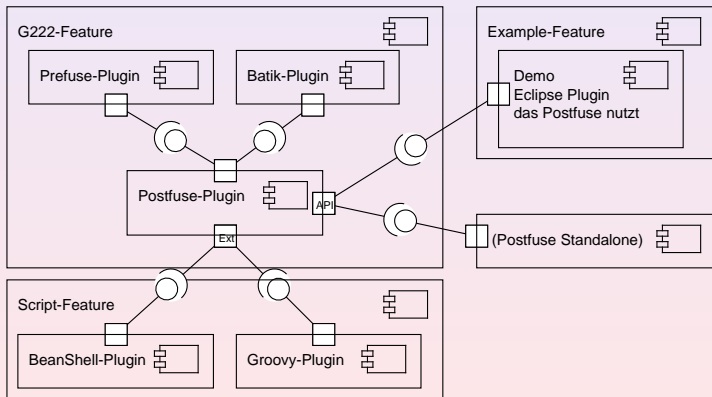
- QS: Funktionalität
- QS: Benutzbarkeit
- freie Lizenz (BSD)
- ...

Gliederung

- 1 Design Überlegungen
- 2 Architektur**
- 3 Use-Cases Realisierungen
- 4 Stand der Dinge

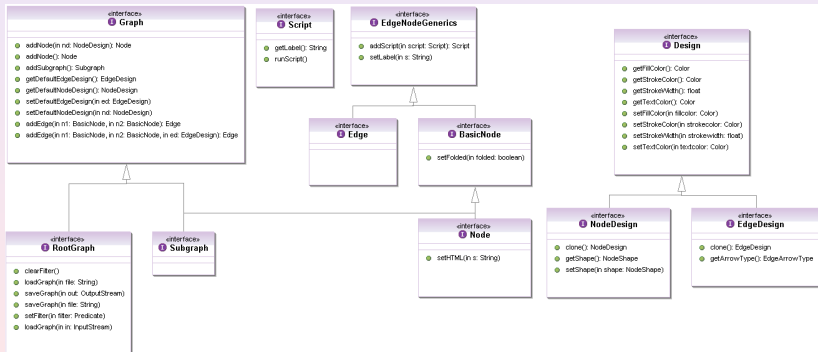
Architektur

Subsysteme und ihre Aufgaben



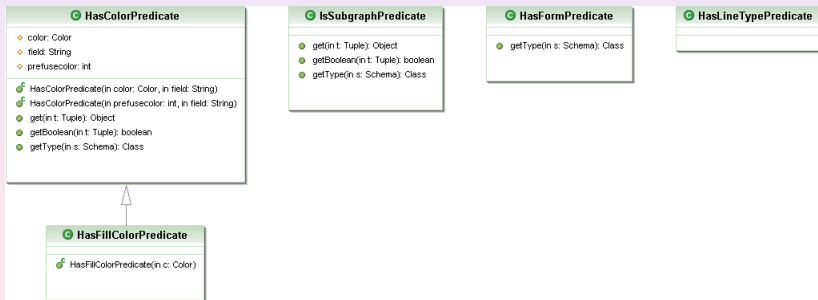
Architektur

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

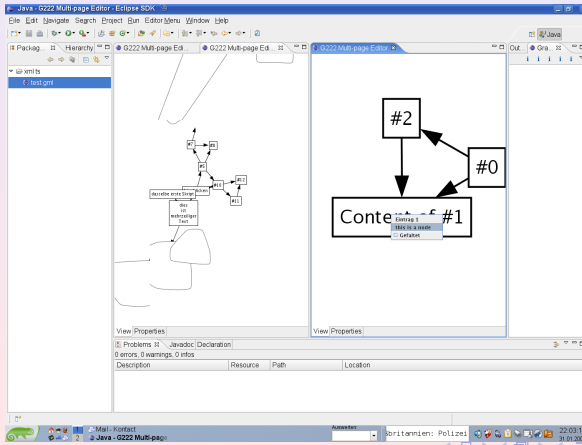
 saveGraph(in g: RootGraph, in file: IFile, in monitor: IProgressMonitor)

 stop(in context: BundleContext)

 start(in context: BundleContext)

Architektur

Die GUI



Architektur

Erweiterungspunkte

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

Architektur

Erweiterungspunkte

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

Architektur

Erweiterungspunkte

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

Architektur

Erweiterungspunkte

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

Architektur

Erweiterungspunkte

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

Architektur

Erweiterungspunkte

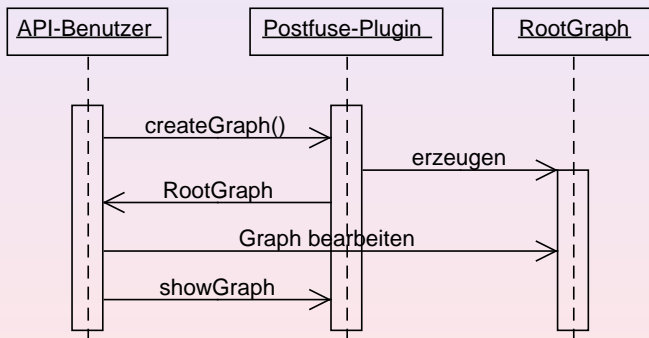
- Andere Skriptsprachen
- Eigene Filter (Prädikate)
- Umgebungen in denen die Skripts ausgeführt werden
- Prefuse Erweiterbarkeit (teilweise) beibehalten
 - Layouter
 - Design der Elemente

Gliederung

- 1 Design Überlegungen
- 2 Architektur
- 3 Use-Cases Realisierungen**
- 4 Stand der Dinge

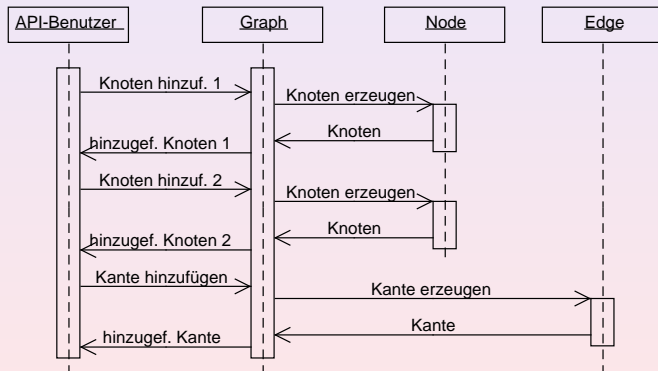
Use-Cases Realisierungen

Graph erstellen und anzeigen



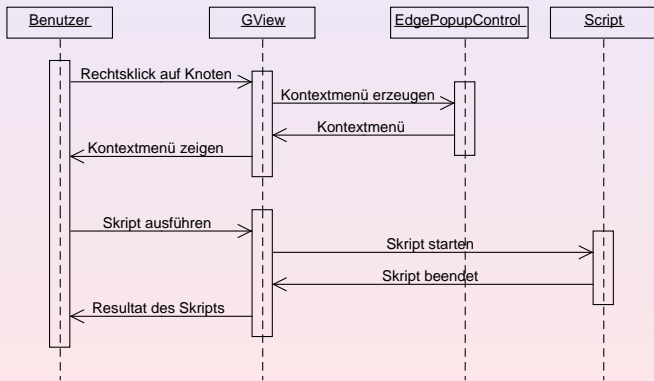
Use-Cases Realisierungen

Knoten & Kanten erstellen



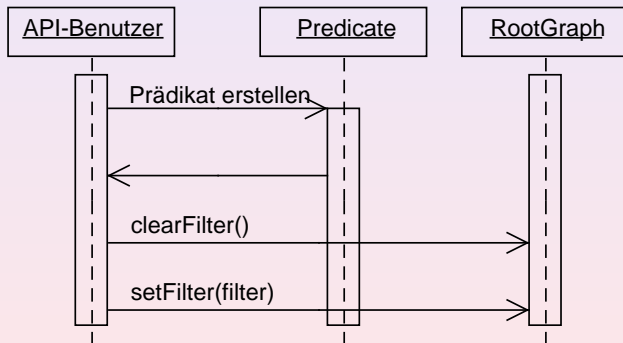
Use-Cases Realisierungen

Skripte



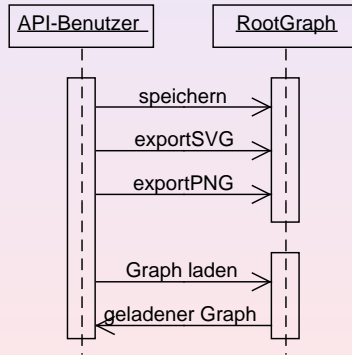
Use-Cases Realisierungen

Filter



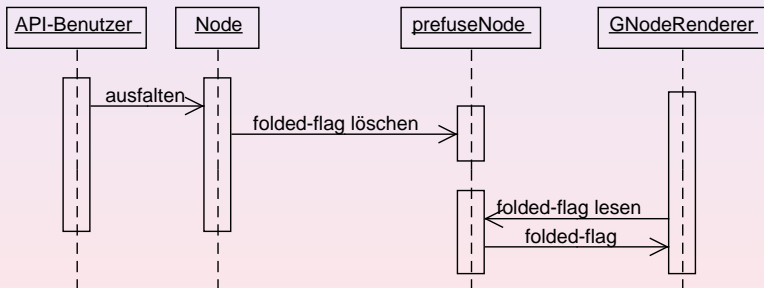
Use-Cases Realisierungen

Laden / Speichern



Use-Cases Realisierungen

Knotenfaltung



Gliederung

- 1 Design Überlegungen
- 2 Architektur
- 3 Use-Cases Realisierungen
- 4 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 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 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 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 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 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 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 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

Was geht noch nicht?

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

Stand der Dinge

Was geht noch nicht?

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

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 & Prototyping
 - 25% Ausarbeitung des Designs
 - 20% Toolsuche und Installation
 - 30% Implementierung
- Schwerpunkt auf die Semesterferien gelegt

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

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

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

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

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

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

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)

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)

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)

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)

Design Überlegungen
Architektur
Use-Cases Realisierungen
Stand der Dinge

Was geht bereits / Demo
Was geht noch nicht
Arbeitszeiten
Weitere Planungen
Fragen

Fragen?

