Dokumentation

$G^{2^{2^2}}$

3. November 2006

http://bp.macrolab.de

Inhaltsverzeichnis

1 Pflichtenheft **3**
1.1 Einleitung 3
1.2 Vision 3
1.3 Ist 4
1.4 Soll 4

1 Pflichtenheft

1.1 Einleitung

Dieses Dokument spiegelt unsere Auffassung des Projektes wieder und ist als Angebot an den Auftraggeber zu verstehen. Durch das Gegenlesen können somit Missverständnisse schon früh im Projekt vermieden werden. Es handelt sich hierbei allerdings nicht um ein statisches Dokument sondern wird fortlaufend mit dem Projekt weitergeschrieben und dem Auftraggeber zum Review vorgelegt.

Die Vision enthält einen groben Überblick darüber, wie wir uns das ideale Produckt am Ende der Entwicklung vorstellen.

Ist gibt Aufschluss darüber, was an Software vorhanden ist. Zudem werden deren Mängel und somit der Grund für diese Entwicklung erläutert.

Soll entspricht der Vision im Detail. Hier sind alle Features einzeln aufgeführt und erläutert.

1.2 Vision

Es soll ein Eclipse-Plugin zur problemlosen Visualisierung von kleinen bis mittleren Graphen entstehen. Die Größe der Graphen sollte dabei etwa 1-50 Knoten umfassen. Der Benutzer soll das Plug-In sowie alle anderen benötigten Plug-Ins einfach und schnell von einer Update-Seite herunterladen können. Dabei sollten keine Abhängigkeiten zu Plug-Ins bestehen, die eine Lizenz erfordern oder anderweitig rechtlich geschützt sind.

Über zwei Schnittstellen sollen Graphen definiert werden können. Zum einen über eine Java-API, so dass einfach und schnell aus anderen Java-Applikationen Graphen erstellt werden können, zum anderen über ausgelesene XML-Dateien. Hierbei ist eine vollständige Interaktion zwischen beiden Erzeugungsmethoden möglich, d.h. über die API erzeugte Graphen können im XML-Format gespeichert werden und aus dem XML-Format erzeugte Graphen können über die API manipuliert werden. Außerdem kann der angezeigte Graph als Bilddatei exportiert werden.

Hauptanwendung ist letztendlich die Visualisierung der Graphen über eine Eclipse-View, die nahtlos in das Eclipse-Konzept eingebettet ist. Die Berechnung des Graphen kann jederzeit vom Benutzer abgebrochen werden. Hierbei werden alle überschaubaren Graphen übersichtlich angzeigt. Den Knoten können Texte beliebiger Länge zugeordnet werden, die sich auf- und zuklappen lassen. Die Möglichkeit Subgraphen und Mehrfachkanten zu verwenden ist auch gegeben. Die, bei Bedarf gerichteten, Kanten sind ebenfalls beschriftbar und zur besseren Unterscheidung farblich markierbar. Ebenso gibt es mehrere Knotentypen, die man verwenden kann. Die Knoten- und Kantentexte lassen sich mit Hilfe von HTML Tags formatieren.

Zusätzlich kann man Skripte an die Knoten und Kanten hängen, die über einen Rechtsklick in den Graphen ausgeführt werden können. Grundbefehle, auf die die Skripte zugreifen können, sind im Plug-In implementiert. Es lassen sich mehrere verschiedene Skriptsprachen verwenden. Große Graphen werden zusätzlich in einem Overview Fenster angezeigt und unterstützen Pan & Zoom Funktionen.

1.3 Ist

Bislang existiert in der Arbeitsgruppe nur ein, nicht trivial zu installierendes, Plug-In zur Visualisierung von Graphen, das keinen Export als Grafik erlaubt. Zudem ist kein XML Dateiformat spezifiziert und es ist nicht in der Lage Funktionen wie Pan & Zoom in der GUI auszuführen. Auch fehlt eine Möglichkeit Skripte einzubinden.

1.4 Soll

Das Soll unserers Projektes kann aus der nachfolgenden Tabelle entnommen werden. Hier sind alle gewünschten Features im Detail beschrieben. Die Prioritäten sind als Schulnoten für das Endergebnis zu verstehen:

5 & 4 sind verpflichtende Features

3 optionale Features, die nache Möglichkeit implementiert werden

2 wünschenswerte Features, deren Implementation im Konzept vorgesehen wird

1 gehört zur ultimativen Ausbaustufe und eigentlich nicht erforderlich

Priorität	Beschreibung
Eclipse Plug-in	
5	Das Eclipse 3.2 Plug-In muss unter Windows und Linux mit denselben Funktionalitäten lauffähig sein. Es dürfen keine Abhängigkeiten zu Tools und Bibliotheken existieren, die nicht nicht unter MacOS X ausführbar sind. Das Plug-In muss unter Java 5 kompilierbar sein.
5	Die Installation erfolgt über eine Eclipse Update-Site, dabei dürfen keine Abhängigkeiten zu Tools existieren, die sich nicht mit Hilfe dieser installieren lassen.
5	Die simultane Anzeige mehrerer Graphen, z.B. nebeneinander, sollte möglich sein.
4	Berechnungen sollen abbrechbar sein, falls sie zu lange dauern
3	Die Anzeige der Graphen sollte als Eclipse View erfolgen.
Visualisierung eines Graphen	
5	Knoten können mit einem mehrzeiligen Text beschriftet werden.
5	Als Zeichensatz für den Text eines Knotens sollte UTF-8 (oder UTF-16) unterstützt werden - die Zeichen, die außerhalb des ASCII Zeichensatzes definiert sind, müssen auch unterstützt werden.
5	Die Form und das Layout der Knoten ist variabel. Es lässen sich Farben und die Dicke des Rahmens variieren.
5	Mehrfachverbindungen zwischen zwei Knoten müssen unterstützt werden und auch visuell erkennbar sein
5	Mehrfachverbindungen eines Knotens mit sich selbst müssen erkennbar sein.
5	Gerichtete und ungerichtete Verbindungen müssen unterstützt werden.

4	Es sollten mindestens fünf verschiedene Typen von Verbindungen unterstützt werden, erkennbar durch unterschiedliche Farbe und Dicke der Linien
4	Es sollte möglich sein, Kanten zu beschriften.
4	Es sollten mindestens drei verschiedene Start- und Endknotentypen verfügbar sein.
3	Unterstützung von Subgraphen.
3	Als Knotentext kann HTML formatierter Text verwendet werden.
3	Zusammengesetzte Knoten, d.h. es sollte möglich sein Text nebeneinander zu setzen.
2	Knoten sollten faltbar sein, d.h. der Benutzer sollte in der Lage sein, Knoten mit viel Text zusammenzufalten und dann nur einen Kurzbezeichner zu sehen.
Interaktion mit dem Graphen	
5	Unterstützung für ein Skript, z.B. in JavaScript, welches an einen Knoten angehängt wird
3	Unterstützung für mehr als ein Skript.
3	Filterung von verschiedenen Verbindungen sollte möglich sein, so dass man Verbindungstypen ausschalten kann
3	Es sollte möglich sein, Basisfunktionalitäten vorzudefinieren, auf die dann in den Skripten zugegriffen werden kann. Zum Beispiel eine Funktion die bei gegebenen Methodennamen und Namen der deklarierenden Klasse einen Editor öffnet und zur entsprechenden Methode navigiert.
3	Es sollte möglich sein angezeigte Graphen als Grafiken exportieren zu können.
2	Ein Overview Fenster das bei größeren Graphen anzeigt wo man sich befindet.
1	Unterstützung für mehr als eine Skriptsprache.
1	Es sollte möglich sein an Verbindungen zwischen zwei Knoten Skripte zu hängen.
Sonstiges Anforderungen	
5	Es dürfen keine offensichtlichen lizenzrechtlichen Beschränkungen existieren, die die freie zur Verfügungstellung des Plug-ins behindern.
5	Eine Webseite inkl. Eclipse Update-Site, welche das Plug-in vorstellt und die Verwendung erklärt.

Literatur

[1] http://www-128.ibm.com/developerworks/opensource/library/os-ecplug/.

[2] Eclipse 3.2 Documentation. http://help.eclipse.org/help32/index.jsp.

[3] PDE Does Plug-ins. http://www.eclipse.org/articles/Article-PDE-does-plugins/PDE-intro.html.

[4] Eclipse.org home. http://www.eclipse.org/, 2006.

[5] Staruml. http://staruml.sourceforge.net/, 2006.

[6] subversion. http://subversion.tigris.org/, 2006.