



Philipps-Universität Marburg
Fachbereich: Mathematik und Informatik
Diplomand: Pawel Stepień
Leitung: Prof. Dr. Gabriele Taentzer
WS 2009/2010

Entwicklung eines Eclipse Plug-In zu Metrikbasierten Qualitätsanalyse von Softwaremodellen

Lastenheft

Inhaltsverzeichnis

1. Zielbestimmungen

2. Produkteinsatz

3. Produktfunktionen

3.1. Spezifikation von neuen Metriken für Metamodelle auf EMF.

3.1.0. Eingabe der Basisdaten.

3.1.1. Beschreibung durch Modelltransformationen.

3.1.2. Zusammensetzung von vorhandenen Spezifikationen.

3.1.3. Speicherung der Metrik.

3.1.4. Generierung des Codes zur Berechnung der Metrik.

3.2. Berechnung der Metriken auf bestimmten Modellen

3.2.1. Laden von Metriken.

3.2.2. Anzeigen von geladenen Metriken.

3.2.3. Metriken für Berechnung konfigurieren

3.2.4. Metriken berechnen.

3.2.5. Berechnungsergebnisse anzeigen.

3.2.6. Berechnungsergebnisse abspeichern.

4. Produktdaten

5. Zusatzanforderungen

5.1. Produktleistungen

5.2. Qualitätsanforderungen

6. Ergänzungen

6.1. Realisierung

6.2. Ausblick auf die nächste Version

1. Zielbestimmungen

Produktname: EMF Metrics

Es soll eine Erweiterung (Plugin) für das Programmierwerkzeug: Eclipse entwickelt werden.

Das Plugin soll Eclipse um ein Werkzeug zur Qualitätsanalyse von auf dem Eclipse Modelling Framework (EMF) basierenden Softwaremodellen erweitern.

Die Qualitätsanalyse soll durch die Berechnung von Modell-Metriken auf den Softwaremodellen geschehen.

Dem Benutzer soll eine Palette an vorgefertigten Metriken für UML2-EMF-Modelle zur Verfügung stehen.

Des Weiteren soll der Benutzer die Möglichkeit haben selber Metriken für EMF basierte Modelle zu spezifizieren. Die Quellcodes für die Berechnung der spezifizierten Metriken sollen automatisch generiert werden.

Das Plugin soll über eine grafische Benutzeroberfläche (GUI) verfügen.

Die GUI soll benutzerfreundliche Schnittstellen für die folgenden Funktionalitäten bereitstellen:

- Spezifikation von neuen Metriken
- Anzeige der zur Verfügung stehenden Metriken
- Konfiguration einer Auswahl von Metriken für ein gegebenes Projekt
- Berechnung der ausgewählten Metriken auf einem Modell
- Anzeige der Berechnungsergebnisse

EMF Henshin Kompatibilität :

Für die Spezifizierung neuer Metriken soll Kompatibilität zu dem EMF Modell-Transformations-Werkzeug: „EMF Henshin“ vorhanden sein.

EMF Henshin unterstützt direkte Transformationen von EMF Modellinstanzen wie auch Generierung von Instanzen einer Zielsprache aus einer gegebenen Instanz einer Quellsprache. Dem Benutzer steht sowohl ein baumbasierter (EMF) als auch ein grafischer (GMF) Editor zur Verfügung.

EMF Henshin ist als ein weiteres Eclipse Plug-In realisiert und kann somit direkt in der gleichen Eclipse Instanz wie EMF Metrics aufgerufen werden.

2. Produkteinsatz

Benutzergruppe: Softwaremodellierer, Reviewer,
Metrikendesigner.

Es wird zwischen zwei Zielgruppen unterschieden.

- Die Softwaremodellierer und Softwarereviewer sollen das Werkzeug zur Berechnung von Metriken benutzen können. Dazu soll eine Auswahl an vorgefertigten Metriken zur Verfügung stehen.
- Die Metrikendesigner sollen mit dem Werkzeug neue Metriken spezifizieren können. Neue Metriken sollen entweder durch die Zusammensetzung der vorhandenen Metriken oder durch die Angabe von Berechnungsregeln spezifiziert werden.

Das Produkt ist als eine Erweiterung von der Eclipse Entwicklungsumgebung konzipiert und kann dementsprechend alleine nicht funktionieren.

Des Weiteren, wird vorausgesetzt dass das EMF Henshin Plugin in die Eclipse Instanz integriert ist.

Das Plugin soll später in ein größeres Modell-Qualitätsanalyse Werkzeugpaket integriert werden. Es soll aber auch als alleinstehende Eclipse Erweiterung funktionieren können.

Das Plugin soll weltweit genutzt werden können und soll deshalb Englisch als Verkehrssprache benutzen.

3. Produktfunktionen

3.1. Spezifikation von neuen Metriken für Metamodelle auf EMF.

Die Spezifikation von neuen Metriken soll über grafische Benutzeroberfläche geschehen. Die benötigten Daten sollen durch entsprechende Eingabeformulare erfasst werden.

Für die Spezifizierung von neuen Metriken sollen zwei Szenarios bereit gestellt werden. Eine neue Metrik soll entweder über die Angabe von Berechnungsregeln oder durch Verweise auf bereits vorhandene Metriken spezifiziert werden. Beide Möglichkeiten setzen die Angabe einer Palette an Basisdaten.

Nachdem alle Informationen für eine neue Metrik erfasst sind, soll die Metrik gespeichert und der Quellcode für ihre Berechnung generiert werden.

3.1.0. Eingabe der Basisdaten.

Bei der Spezifikation einer neuen Metrik sollen die folgenden Basisaten erfasst werden:

- Metriken-Projekt in das der Code zur Berechnung der neuen Metrik generiert werden soll.
- Name der Metrik.
- Beschreibung der Metrik.
- Metamodell für das die Metrik angewendet werden können soll.
- Kontext in dem die Metrik angewendet werden können soll
- Wertebereich für das Ergebnis der Berechnung der Metrik

3.1.1. Beschreibung durch Modelltransformationen.

Mit Hilfe von EMF Henshin sollen Transformationsregeln für die Metriken erstellt werden. Diese sollen dann bei der Spezifikation angegeben und in das System eingebunden werden. Anhand von den Transformationsregeln sollen schließlich die Metriken berechnet werden.

Es kann vorkommen dass eine Transformationsregel nicht ausreicht um die gewünschte Metrik zu berechnen. Aus diesem Grund soll es möglich sein, Pakete zu definieren, die aus bis zu drei Transformationen bestehen: eine Transformation die das Modell für die Berechnung vorbereitet, eine Transformation zur Berechnung der Metrik und eine abschließende Transformation die das Modell wieder in den Anfangszustand bringt.

Zusätzlich zu den Basisdaten sollen in diesem Szenario die folgenden Informationen erfasst werden:

- Verweise auf die Dateien mit den Modelltransformationen.

3.1.2. Zusammensetzung von vorhandenen Spezifikationen.

In diesem Szenario soll erstmal eine Liste mit allen verfügbaren Metriken angezeigt werden. Der Benutzer soll aus dieser Liste, die für die Zusammensetzung gewünschten Metriken, auswählen können. Des weiteren soll eine mathematische Operation zur Verbindung von den markierten Metriken ausgewählt werden. Die Operation soll ebenso aus einer Liste aller Verfügbaren Operationen ausgewählt werden können.

Das Ergebnis der neu spezifizierten Metrik soll aus den Ergebnissen der jeweiligen Metriken unter der Ausführung der mathematischen Operation bestehen.

Zusätzlich zu den Basisdaten sollen in diesem Szenario die folgenden Informationen erfasst werden:

- Verweis auf die erste Metrik
- Verweis auf die zweite Metrik
- Definition der mathematischen Bindeoperation

3.1.3. Speicherung der Metrik.

Nachdem die Metrik erstellt worden ist soll sie auf der Festplatte abgespeichert werden, so dass sie auch nach dem Schließen der Eclipse Instanz beständig bleibt und für spätere Sessionen zur Verfügung steht.

Das Speichern der neuen Metrik soll durch den erfolgreichen Abschluss des Spezifikationsprozesses automatisch initialisiert werden.

Es wird erfordert, dass die gespeicherte Metrik beim nächsten Start des Plugin automatisch erkannt wird sofern das Zielprojekt noch vorhanden ist. Dies soll möglichst über interne Strukturen von Eclipse sichergestellt werden.

3.1.4. Generierung des Codes zur Berechnung der Metrik.

Der Code zur Berechnung der neu erstellten Metrik soll bei der Speicherung der Metrik automatisch generiert werden. Dafür sollen keine weiteren Interaktionen mit dem Benutzer erforderlich sein.

Die Datei mit dem Code soll in das aus den Basistaten entnommene Verzeichnis gespeichert werden.

Es soll eine feste Verknüpfung zwischen der Datei mit dem Berechnungscode und den dazugehörigen Henshin Dateien erstellt werden. Es soll aber möglich sein das Projektverzeichnis in dem sich die beiden Daten befinden zu ändern ohne dass diese Änderung eine Auswirkung auf die Funktionalität des Codes hat.

Falls nötig sollen im Zielprojekt neue Klassenabhängigkeiten gesetzt werden, so dass der neu erstellte Code ohne Probleme kompiliert werden kann.

3.2. Berechnung der Metriken auf bestimmten Modellen

Die Berechnung der Metriken auf einem bestimmten Modell soll durch die GUI initialisiert werden können. Die Liste der zu berechneten Metriken soll vor dem Start der Berechnung über die Benutzeroberfläche auf eine von dem Benutzer definierte Auswahl eingeschränkt werden können.

Nach der erfolgreichen Ausführung der Berechnungen, sollen die jeweiligen Ergebnisse in der GUI angezeigt werden. Es soll möglich sein die Ergebnisse abzuspeichern.

Des Weiteren soll es möglich sein die Liste der Ergebnisse zwischen den Berechnungen zu löschen.

3.2.1. Laden von Metriken.

Die Metriken sollen bei der Programminitialisierung geladen werden. Dieser Prozess soll automatisch ausgeführt werden. Dafür sollen keine Interaktionen mit dem Benutzer erforderlich sein.

Die Liste der bereitstehenden Metriken soll aus allen Metriken bestehen, die in Projekten spezifiziert worden sind, die bei der Programminitialisierung vorhanden waren.

Die Liste der Metriken soll jedes mal bei der Initialisierung einer neuen Session neu geladen werden um sicher zu stellen dass die aktuell ist. Nach der Terminierung des Plugins soll sie wieder gelöscht werden.

3.2.2. Anzeigen von geladenen Metriken.

Es soll möglich sein alle geladenen Metriken über die GUI anzuzeigen. Die Anzeige soll aus den folgenden Feldern bestehen:

- Name
- Beschreibung
- Kontext
- Metamodell

Die Metriken sollen nach Metamodellen gruppiert werden. Des Weiteren, soll eine Untergruppierung nach dem Kontext der Metriken erfolgen.

3.2.3. Metriken für Berechnung konfigurieren

In der Metrik-Anzeige (3.2.2) soll es möglich sein die gewünschten Metriken zu markieren, um die so entstandene Auswahl zu speichern. Es soll möglich sein für jedes Projekt eine andere Auswahl von Metriken zu haben.

Beim öffnen der Metrik-Anzeige (3.2.2) für ein bestimmtes Projekt sollen in der Liste aller verfügbaren Metriken die ausgewählten Metriken markiert werden.

Es soll auch möglich sein diese Markierungen über die Metrik-Anzeige (3.2.2) wieder zu entfernen. Die Anzeige soll zusätzlich zu den in 3.2.2 definierten Feldern noch ein zusätzliches Feld für die Konfiguration besitzen.

3.2.4. Metriken berechnen.

Die Berechnung der Metriken soll aus der GUI gestartet werden können. Nachdem der Benutzer ein konkretes Objekt eines Modells ausgewählt hat, soll er die Möglichkeit haben Metriken auf dem Objekt zu berechnen.

Die Liste der Metriken für die Berechnung soll anhand der in 3.2.3 ausgewählten Metriken erstellt werden. Die zu berechnenden Metriken sollen jedoch alle dem Metamodell wie auch dem Kontext des ausgewählten Objektes entsprechen. Die Liste aus 3.2.3 soll für jedes ausgewählte Objekt entsprechend gefiltert werden.

Es sollen nur die Metriken berechnet werden, die für das richtige Metamodell spezifiziert worden sind und die für den Kontext des Objektes anwendbar sind.

3.2.5. Berechnungsergebnisse anzeigen.

Die Ergebnisse der Berechnungen aus 3.2.4 sollen in der GUI angezeigt werden. Die Anzeige soll aus den folgenden Feldern bestehen:

- Datum und genaue Uhrzeit der Berechnung
- Der konkrete Kontext in dem die Metrik berechnet worden ist.
- Name der Metrik die berechnet worden ist
- Beschreibung der Metrik.
- Ergebnis der Berechnung

Für weitere Berechnungen sollen die Ergebnisse zu dieser Liste hinzugefügt werden. Dabei sollen die schon vorhandenen Einträge nicht gelöscht werden.

Es soll möglich sein zwischen einzelnen Berechnungen die Liste der Ergebnisse zu löschen.

3.2.6. Berechnungsergebnisse abspeichern.

Es soll möglich sein die gesamten Berechnungsergebnisse zu speichern. Der Inhalt der Anzeige aus 3.2.5 soll auf der Festplatte gespeichert werden können. Dabei sollen Daten aus allen in 3.2.5 angegebenen Feldern gespeichert werden.

Der Befehl für die Speicherung soll aus der GUI zugänglich sein.

Die Ergebnisse sollen in eine separate Datei, gespeichert werden. Die Datei soll über ein gut erweiterbares und möglichst unkompliziertes Format verfügen.

4. Produktdaten

4.1. Die Metriken - sollen separat - unabhängig vom Speicherort des EMF Metrics Plugins - gespeichert werden. Sie sollen Plugin Initialisierung leicht geladen werden können.

4.2. Die Dateien mit den Berechnungscodes sowie die **Henshin Transformationsdateien** - sollen mit einander fest verknüpft und in gleichem Zielprojekt gespeichert werden.

4.3. Die Liste aller verfügbaren Metriken - soll nur vorübergehend für jede Session separat gespeichert werden.

4.4. Die projektspezifische Auswahl der Metriken (4.2.3) - soll in einer separaten Datei ebenfalls in gleichem Projekt gespeichert werden.

4.5. Die Berechnungsergebnisse - sollen extern gespeichert werden können. Es muss in einer Datei im Textformat gespeichert werden können. Die Datei soll über ein gut erweiterbares und möglichst unkompliziertes Format verfügen.

5. Zusatzanforderungen

5.1. Produktleistungen

5.1.1 Benutzerfreundlichkeit – Die Funktionalitäten sollen sich der GUI leicht und möglichst intuitiv zu finden sein.

5.1.2. Akkumulation - Bei fehlererzeugenden Eingaben soll der Benutzer als Fehlermeldung eine Auflistung aller eingegebenen Fehler.

5.1.3. Toleranz - Bei fehlerhaften Eingaben muss der Benutzer die Möglichkeit haben, eine Korrektur der Eingabedaten vorzunehmen, ohne Eingaben wiederholt eingeben zu müssen.

5.1.4. Konsistenz – Alle gespeicherten Daten sollen zwischen verschiedenen Sessionen unverändert bleiben.

5.1.5. Korrektheit – Alle spezifizierten Metriken sollen den Spezifikationen entsprechende Werte liefern.

5.2. Qualitätsanforderungen

6.1.1. Es wird ein besonders großer Wert auf ausführliche Dokumentation gelegt.

6.1.2. Es wird eine Modellbasierte Vorgehensweise erfordert.

6.1.3. Gute Erweiterbarkeit sollte gewährleistet sein.

6. Ergänzung

6.1. Realisierung

Als Programmiersprache wird Java verwendet.

Das Plug-In soll in die Eclipse Galileo modelling version eingebunden werden.

Als Modell-Transformations-Werkzeug soll EMF Henshin benutzt werden.

Innerhalb von 6 monaten

6.2. Ausblick auf die nächste Version

Dieses System ist hier nur mit den Grundanforderungen angegeben und wird bei Erfolg mit weiteren Funktionalitäten versehen:

- Es werden weitere Metriken spezifiziert und zu der integrierten Liste hinzugefügt.
- Die GUI wird erweitert
- Es wird auf die Ergebnisdatei zugegriffen und der Format wird erweitert
- Das Werkzeug wird in ein größeres Projekt integriert