

Pflichtenheft zum Editor für Eclipse GMF der Gruppe HK-07-1

Inhaltsverzeichnis

1. Zielbestimmung	2
1.1 Musskriterien.....	2
1.2 Abgrenzungskriterien.....	2
2. Produkteinsatz.....	2
2.1 Anwendungsbereiche	2
2.2 Zielgruppen	2
2.3 Betriebsbedingungen.....	2
3. Produktumgebung.....	3
3.1 Software	3
3.2 Produkt-Schnittstellen.....	3
3.3 Produktumwelt	3
4. Produktfunktionen	3
5. Produktdaten	4
6. Produktleistungen.....	4
7. Benutzungsoberfläche	5
8. Qualitätsanforderungen.....	5
9. Nichtfunktionale Anforderungen	5
10. Technische Produktumgebungen.....	6
10.1 Software	6
10.3 Orgware.....	6
10.4 Produkt Schnittstellen	6
11. Spezielle Anforderungen an die Entwicklungsumgebung	6
11.1 Software	6
11.2 Hardware	6
11.3 Orgware.....	6
11.4 Produkt Schnittstellen	6
12. Gliederungen in Teilprojekte	7
13. Ergänzungen.....	7
14. Glossar.....	7

1. Zielbestimmung

Ziel ist die Entwicklung eines grafischen Editors, der als Eclipse Plug-In auf Basis des GMF aufsetzt. Mit diesem Editor ist es möglich, selbstdefinierte Metamodelle und Modellierungssprachen grafisch umzusetzen und so die Modellierung von Software zu beschreiben.

1.1 Musskriterien

- Grafische Objekte anlegen/ändern/löschen
- Relationen zwischen Objekten anlegen/ändern/löschen
- Erstellung der gmfgraph, -tool und -map Modelle
- Speichern/Löschen des Modells
- Aus dem definierten Modell soll ein modell-spezifischer Editor generiert werden
- Toolbar/Palette des Editors definieren

1.2 Abgrenzungskriterien

- Aus dem vom Benutzer definierten Modell wird kein Quellcode generiert

2. Produkteinsatz

Das Produkt dient der Entwicklung problemspezifischer Modelle, sowie deren Einbindung in die Eclipse Entwicklungsumgebung.

2.1 Anwendungsbereiche

Mit Hilfe des Produktes ist es dem Anwender möglich Software zu modellieren. Der Editor ermöglicht es, selbstdefinierte Metamodelle und Modellierungssprachen grafisch darzustellen, um damit Software beschreiben zu können. Das Produkt erleichtert also die Modellierung von Software im Großen und durch besondere Intuitivität speziell auch im kleinen Rahmen. Dabei überzeugt es durch Benutzbarkeit und Visualisierung.

2.2 Zielgruppen

Zielgruppe sind erfahrene Modellierer, Entwickler und Implementierer, die das Produkt in umfangreichen Softwareprojekten einsetzen. Weiterhin können auch fortgeschrittene Nutzer das Programm betreiben. Zu diesen unter anderen Zählen Studenten und andere Interessierte.

2.3 Betriebsbedingungen

Das Produkt ist auf allen Java-fähigen Computern betriebsfähig.

3. Produktumgebung

3.1 Software

Da das Produkt als Plug-In für Eclipse entwickelt wird, ist zu verlangen, dass auf der Zielumgebung Eclipse mit GMF vorinstalliert ist.

3.2 Produkt-Schnittstellen

Das Produkt wird in die Eclipse Entwicklungsumgebung integriert.

3.3 Produktumwelt

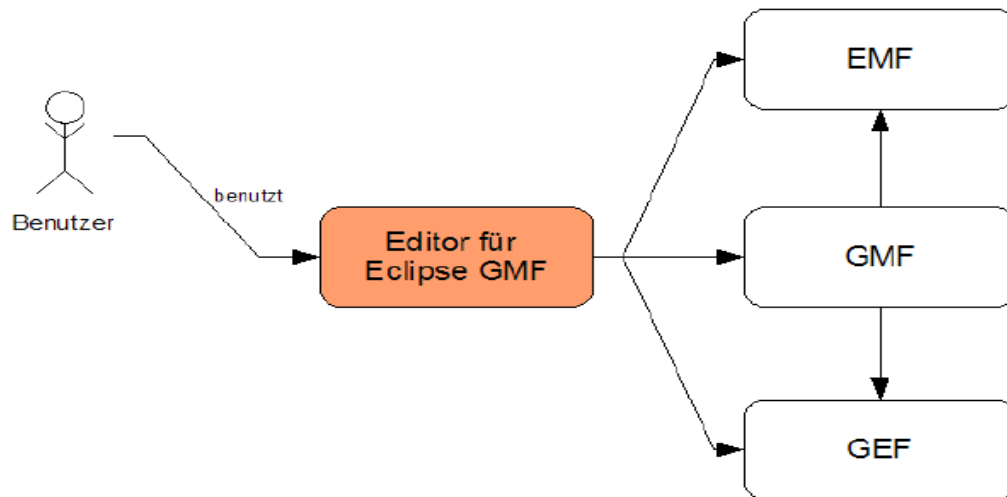


Abbildung 1: Umweltdiagramm

4. Produktfunktionen

- /F010/ Geschäftsprozess: Grafische Oberfläche erstellen
Erstellt eine neue Arbeitsfläche, auf der grafische Objekte positioniert werden können.
 - /F020/ Geschäftsprozess: Grafische Objekte erstellen
Erstellt ein neues grafisches Objekt auf der Oberfläche.
 - /F030/ Geschäftsprozess: Grafische Objekte verknüpfen
Verknüpft zwei vorhandene Objekte.
 - /F040/ Geschäftsprozess: Grafische Objekte benennen/umbenennen
Weist einem grafischen Objekt einen neuen Namen zu oder ändert den vorhandenen.
 - /F050/ Geschäftsprozess: Größe von Objekte ändern
Verändert die Maße eines Objektes anhand eines gegebenen Koordinatensystems (Skalierung von Objekten).
-

- /F060/ Geschäftsprozess: Verknüpfungen benennen/umbenennen
Weist einer Verbindung einen neuen Namen zu oder ändert den vorhandenen.
- /F070/ Geschäftsprozess: Übersetzen eines Modells in GMF-Daten
Das rohe Modell auf der grafischen Oberfläche wird in ein für GMF lesbares Datenformat übersetzt.
- /F080/ Geschäftsprozess: Löschen von Objekten und Verknüpfungen
Löscht ein Objekt oder eine Verbindung von der grafischen Oberfläche
- /F090/ Geschäftsprozess: Speichern eines erstellten Modells
Veranlasst das System, ein erstelltes Modell abzuspeichern.
- /F100/ Geschäftsprozess: Löschen eines erstellten Modells
Lässt das System ein erstelltes Modell verwerfen.
- /F110/ Geschäftsprozess: Attribut hinzufügen
Fügt einem Objekt ein Attribut hinzu.
- /F120/ Geschäftsprozess: Attribute entfernen
Entfernt ein Attribut von einem Objekt.

5. Produktdaten

- /D10/ ECore-Modell muss im XML-Format abgelegt werden.
- /D20/ GMF-Modelle müssen im XML-Format abgelegt werden.

6. Produktleistungen

- /L10/ Im Abgleich mit /LD10/ und /LD20/wird ein universell lesbares, wellformed XML-Datendokument ausgegeben.
- /L20/ Im Abgleich mit allen geometrischen Funktionen ist die Darstellung selbiger in Echtzeit umzusetzen. Dabei ist eine Verzögerung von 1s akzeptabel.
- /L30/ In Betracht von u.a. /LF20/ und /LD10/ist die gleichzeitige Darstellung grafischer Objekte auf eine maximale Anzahl von 500 zu begrenzen
- /L40/ Es stehen nicht mehr als 10 verschiedene grafische Elemente zur Verfügung

7. Benutzungsoberfläche

Da das Produkt als Eclipse Plug-In entwickelt wird, ist die grundsätzliche Visualisierung von Eclipse vorgegeben.

Alle Formen, die vom Benutzer zum editieren verwendet werden, sind ebenfalls standartmäßig aus der Eclipse Bibliothek. Die Visualisierung des Editors erfolgt auf Basis von Eclipse und kann durch einzelne Views ergänzt werden, die die Sichtweite des Programms erhöhen.

/B10/ Die Oberfläche soll durch die Maus gesteuert werden.

/B20/ Die Bedienung des Editors ist selbstklärend, wird aber genauer im Handbuch beschrieben.

/B30/ Die Oberfläche, sowie deren Elemente, werden verständlich und intuitiv gestaltet.

8. Qualitätsanforderungen

Produktqualität	Sehr gut	Gut	Normal	Nicht relevant
Funktionalität				
Angemessenheit			X	
Richtigkeit		X		
Interoperabilität		X		
Ordnungsmäßigkeit			X	
Sicherheit				X
Zuverlässigkeit				
Reife			X	
Fehlertoleranz			X	
Wiederherstellbarkeit			X	
Benutzbarkeit				
Verständlichkeit		X		
Erlernbarkeit	X			
Bedienbarkeit	X			
Effizienz				
Zeitverhalten				X
Verbrauchsverhalten				X
Änderbarkeit				
Analysierbarkeit			X	
Modifizierbarkeit		X		
Stabilität		X		
Prüfbarkeit			X	
Übertragbarkeit				
Anpassbarkeit			X	
Installierbarkeit			X	
Konformität		X		
Austauschbarkeit		X		

9. Nichtfunktionale Anforderungen

Für die Strukturierung und Richtigkeit der Modelle ist allein der Benutzer verantwortlich.

10. Technische Produktumgebungen

10.1 Software

Für die Entwicklung des Produkts wird die Eclipse IDE selbst benötigt. Auf dieser ist GMF installiert, auf dem der Editor aufsetzt. Zum implementieren des Quellcodes benutzen wir das JAVA SDK 5, inklusive JAVADOC. Ein entsprechender Compiler ist dort bereits vorinstalliert.

10.2 Hardware

Es ist voraus zu setzen, dass die Hardware des Zielsystems die aktuellen Versionen unterstützt. Dazu wird ein leistungsfähiger Personal Computer benötigt, der den aktuellen Leistungen gerecht wird.

10.3 Orgware

Organisatorische Randbedingungen werden zum Betrieb des Produktes nicht benötigt.

10.4 Produkt Schnittstellen

Das Produkt wird innerhalb einer Eclipse-Europa Version, speziell Version 3.3 entwickelt. Der Editor ist als Plug-In anzusehen, der in Eclipse auf Basis von GMF ausgeführt wird. Dazu verwenden wir GMF Version 1.03.

11. Spezielle Anforderungen an die Entwicklungsumgebung

11.1 Software

Zur Entwicklung benutzen wir Eclipse 3.3, sowie GMF 1.03.

11.2 Hardware

Die benutzte Hardware entspricht den aktuellen Leistungsmerkmalen.

11.3 Orgware

Orgware wird nicht benötigt.

11.4 Produkt Schnittstellen

Das Produkt ist ein Eclipse Plug-In. Sofern wird es innerhalb Eclipse ausgeführt und bearbeitet.

12. Gliederungen in Teilprojekte

Der Editor ist ein einzelnes Plug-In, sofern existieren keine Teilprodukte.

13. Ergänzungen

-/-

14. Glossar

Arbeitsfläche

Die Umgebung des Editors ist die Arbeitsoberfläche. Auf ihr werden Benutzeraktionen ausgeführt.

Constraint

Constraints sind Beschränkungen innerhalb des Modells, die durch Drittsprachen definiert werden können.

DSL

Eine Domain Specific Language ist eine Modellierungssprache, die im Gegensatz zu der sehr allgemeinen Unified Modeling Language (UML) auf ein spezielles Einsatzgebiet/Geschäftsfeld abgestimmt ist.

EClass

Eine EClass ist eine Instanz einer Klasse aus dem Paket org.eclipse.emf.ecore EClass. Eine EClass ist damit das Gegenstück zu java.lang.Class. Die EClass dient also der Beschreibung von Klassen. Sie enthält EAttributes, EOperations und EReferences.

Eclipse

Eclipse ist ein Open-Source Framework zur Entwicklung von Software. Weitestgehend wird Eclipse zur Entwicklung von Java-Applikationen verwendet. Durch seine offene Plug-In Struktur kann Eclipse jederzeit funktional erweitert werden.

Ecore

Ein Meta-MetaModell zum definieren von Meta-Modellen.

EMF

Hinter dem Eclipse Modeling Framework (EMF) verbirgt sich ein Framework, das zur Entwicklung von Tools für die Eclipse-IDE genutzt wird. EMF enthält zusätzlich eine Bibliothek zum Erstellen und Modifizieren von XML-Schemas (XSD) und für Service Data Objects (SDO).

Framework

Ein Framework ist eine Rahmenstruktur die mit dem Software-Paket und der Softwarekomponente in Beziehung steht. Es wird im Rahmen der objektorientierten Softwareentwicklung für komponentenbasierte Entwicklungsansätze verwendet. Das Framework gibt in der Regel eine Anwendungsarchitektur vor.

Form

Objekte/ Verknüpfungen können eine unterschiedliche grafische Darstellung besitzen. Diese werden Formen genannt.

GEF

Das Graphical Editor Framework (GEF) stellt ein Framework dar, über das graphische Editoren für Eclipse programmiert werden können. Das GEF ist nur für den grafischen Editor zuständig, die konkrete Funktionalität eines Tools muss an anderer Stelle programmiert sein (z.B. über EMF).

GMF

Das Graphical Modeling Framework (GMF) ist ein Framework, welches sich zum Ziel gesetzt hat funktionsfähige graphische, Eclipse-basierte Editoren für selbst definierte EMF basierte Metamodelle zu generieren. GMF beinhaltet unter anderem folgende Modelle:

- **gmfgraph**
Das gmfgraph Modell beschreibt die grafischen Formen des Editors. Dazu zählen unter anderem Kreise, Ellipsen, Rechtecke.
- **gmftool**
Das Modell gmftool dient zur Erstellung von Toolbars innerhalb des GMF.
- **gmfmap**
Um die Elemente aus gmfgraph und gmftool zu verbinden, wird zum Mapping gmfmap benutzt.

GMF-Modelle

*Zu verstehen sind hier die Modelle gmfgraph, gmfmap und gmftool. Siehe auch **GMF**.*

Grafische Oberfläche

*Eine grafische Oberfläche ist eine integrierte Zeichenoberfläche von GMF. Auf ihr werden die grafischen Objekte gezeichnet. Die grafische Oberfläche wird auch als **Canvas** bezeichnet.*

JUnit

JUnit ist ein Framework das automatisierte Tests des Systems erlaubt.

MetaEdit+

MetaEdit+ wird zur Definition von Metamodellen verwendet und generiert gleichzeitig einen zugehörigen Editor. MetaEdit+ wird von MetaCase vertrieben.

- **MetaCase**
MetaCase ist einer der wichtigsten Anbieter von Produkten zur domänenspezifischen Modellierung.

Metamodell

Modelle, die beschreiben, wie Modelle gebaut werden nennt man Meta-Modelle. Ein Metamodell enthält Methoden, die Syntax und Semantik der Modelle definieren. Diese Methoden selbst sind wieder reale Methoden, deswegen liegt es nahe, diese Methoden auch durch Modelle zu beschreiben.

Metamodellierungssprache

Eine Metamodellierungssprache die in ihrer Abstraktionsebene höher steht als Modellierungssprachen und wird deshalb zu deren Entwicklung genutzt.

Objekt

Unter einem Objekt sind die verschiedenen grafischen Formen zu verstehen. Diese werden als Node angegeben.

MVC-Architektur

MVC ist ein weitverbreitetes Architekturkonzept. Die Aufgabe dieser Architektur besteht darin die Bestandteile Modell, View und Controller von einander zu trennen. Dabei repräsentiert das Modell die bestehenden Daten, die über das View abgebildet werden und vom Controller manipuliert werden.

Palette

Ein anderes Wort für die Toolbar des Editors. Aus ihr werden Objekte und Verknüpfungen gewählt.

Plattform

Die Plattform ist die Zielumgebung, wie das Betriebssystem oder virtuelle Maschine, auf der das entstehende Produkt benutzt wird.

Plug-In

Ein Plug-In ist ein Computerprogramm, welches in ein anderes Software-Produkt integriert wird. Es ergänzt dabei die Software. Anders als ein Add-on stellt es jedoch eine eigenständige Software dar.

Service Data Objects (SDO)

SDO ist eine Spezifikation für ein herstellerunabhängiges Framework zum einheitlichen Datenzugriff, das eine gute Konnektivität zu XML bereitstellt. Es ermöglicht damit einen uniformen Datenzugriff auf heterogene Datenquellen.

Syntax

Hier ist zwischen konkreter und abstrakter Syntax zu unterscheiden.

- **Konkrete Syntax**
Gemeint ist die Syntax, wie sie von einer Sprache definiert wurde.
- **Abstrakte Syntax**
In der abstrakten Syntax werden die grundlegenden Datenstrukturen definiert. Diese werden in der konkreten Syntax verwendet.

Verknüpfungen/Relationen

Verknüpfungen sind Verbindungen/Assoziationen zwischen Objekten.

Verknüpfungsart

Verknüpfungen können unterschiedliche Deutungen haben. Diese werden Arten genannt.

Wizard

Der Begriff Wizard bezeichnet eine Oberfläche, mittels derer ein Anwender durch mehrere Dialoge, für eine ergonomische Dateneingabe geführt wird.

XML

Die Extensible Markup Language bezeichnet einen Standard zur Erstellung von strukturierten Daten in Form eines Textdokuments. Die Dokumente haben die Form einer Baumstruktur, die vom World Wide Web Consortium (W3C) definiert wurde. Die Regeln für den Aufbau solcher Dokumente werden von XML vorgegeben.