Bachelor-Thesis

an der
Ostfalia Hochschule für angewandte Wissenschaften

im Fachbereich Informatik
im Studiengang IT-Management

Entwicklung einer domönspezifischen Sprache zur Unterstützung der Angebotserstellung

Autor: Stephan Elvers
Matrikel-Nr.: 70382189

Erstprüfer: Prof. Dr. Ina Schiering
Zweitprüfer: B.Sc Christopher Klein

Abgabe am: 0. Mai 2019

Inhaltsverzeichnis

1 Abstrakt

In Softwareprojekten wird der Großteil des Quellcodes von Hand geschrieben, obwohl sich viele der notwendigen Artefakte aus vorhandenen Informationen automatisiert generieren ließen. Die Generierung von Quellcode reduziert die Fehleranfälligkeit innerhalb der zu entwicklenden Anwendung und sorgt dafür, dass mehr Zeit in die Implementierung der Geschäftslogik investiert werden kann.

In dieser Bachelorarbeit wird untersucht.....

2 Einleitung

"Write Code That Writes Codes"[HT06] *Kursiv*

2.1 Aufgabenstellung

unterkapitel

Das im Rahmen dieser Bachelorarbeit entstehende Eclipse Plug-in soll dabei so entwickelt werden, dass neue Generatoren für Code-Fragmente einfach integriert werden können.

2.2 Motivation

Auslöser für diese Bachelor-Arbeit ist ein abgeschlossenes Software-Projekt gewesen, in dem bereits Xtext im kleinen Rahmen verwendet wurde. Die dabei erstellte domänenspezifische Sprache (*Domain Specific Language*, *DSL*) beschrieb die Domänen-Objekte mit ihren Attributen, die mit Hilfe von Xtext in C#-Quellcode umgewandelt wurden. Die DSL wurde zwar nur für grundlegende Transformationen benutzt, dennoch lag der dadurch entstandene Mehrwert auf der Hand. Es entstand eine einheitliche Basis für Dokumentation und Quellcode. Die Zeit, die für alltägliche Programmierarbeiten wie dem Implementieren von Entitäten oder der Datenzugriffsschicht anfiel, wurde auf ein Minimum verringert.

2.3 Vorgehen

Kapitel **??** ¡- Verweis auf anderes Kapitel

Liste

- Pfadangaben und Dateinamen werden in kursiver Schrift dargestellt, z.B. *src/plugin.xml*
- Quellcode wird in Verbatim dargestellt, z.B. `addComponent(...)` ¡– Fett/-Verbatim
- ....

3 Ausgangssituation

3.1 Kontext

3.2 Aktueller Entwicklungsprozess

Im Folgenden soll ein Einblick in den aktuell vorhandenen Softwareentwicklungsprozess bei der NeosIT GmbH gegeben werden. Eine detaillierte Beschreibung würde den Rahmen dieser Arbeit sprengen.

3.2.1 Requirements Engineering

unterunterkapitel
[1]rt. Tabelle 3.1

Komponente	C#	Java
Logging	NLog oder log4net	log4j
Dependency Injection	Unity oder Spring.NET	Spring oder Guice
MVC	ASP.NET MVC 2/3	Spring MVC
Scheduling	Quartz.NET	Quartz
Messaging Queue	ActiveMQ	ActiveMQ
AOP	Unity	AspectJ
Testing	Xunit	JUnit

Tabelle 3.1: Eingesetzte Frameworks

3.2.2 Generierung eines Mockups

```
ALTER TABLE xyz ADD COLUMN nachname char(255);
```

ohne ¡— Fett
..... `Module` implementiere ... ODER .....Variablen `artefakt` gespeichert werden.... ¡—- Verbatism im fließtext

3.2.3 Erstellung einer neuen DSL

- Der Benutzer erstellt innerhalb der Eclipse-Umgebung ein neues Xtext-DSL-Projekt. Dabei muss er den Namen der DSL *$dsl-name*, sowie den Paketnamen der DSL *$paket.name* definieren.

[1]DAL bzw. DAO-Entwurfsmuster

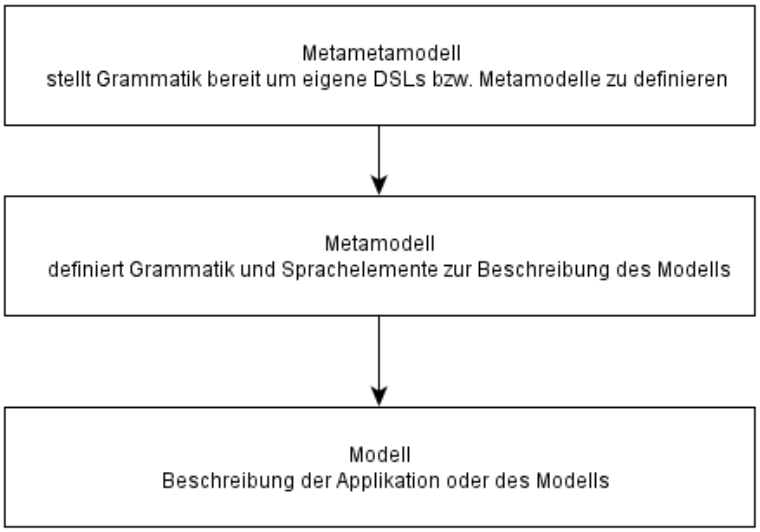

Abbildung 3.1: Modell, Metamodell und Metametamodell

- Das Eclipse Plug-in des Xtext-Frameworks generiert daraufhin automatisch drei Eclipse Plug-ins:
  - *$paket.name*: Grundgerüst für das DSL-Backend. Dieses enthält den Parser, Lexer, Formatter, Metamodell, Scoping- und Validation-Provider. Der generierte Quellcode ist standardmäßig nicht abhängig von der Eclipse-Laufzeitumgebung und kann auch in Konsolen- oder Webanwendungen wiederverwendet werden.
  - *$paket.name*.test: Grundgerüst für das Ausführen von Unittests
  - *$paket.name*.ui: Grundgerüst für das User Interface. Dies beinhaltet unter anderem Content Assistents, Quickfixing und Outline Views. Der generierte Quellcode hängt dabei

src-gen erzeugt.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere, dass ich alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, und dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

Ort, Datum	Unterschrift

Abkürzungsverzeichnis

API Application Programming Interface

CRUD Create, Read, Update, Delete

CSS Cascading Stylesheet

CSV Comma-separated values

DAL Data Access Layer

DAO Data Access Object

DBMS Datenbank Management System

DDL Data Definition Language

DET Data Element Type

DSL Domain Specific Language

ELF External Logical File

EMF Eclipse Modeling Framework

FTR File Type Reference

FPA Function Point-Analyse

HTML Hypertext Markup Language

IDE Integrated Development Environment

ILF Internal Logical File

JDT Java Development Toolkit

JPA Java Persistence API

JVM Java Virtual Machine

MDA Model Driven Architecture

MWE Model Workflow Engine

PDE Plug-in Development Environment

PDT PHP Development Toolkit

POJO Plain Old Java Object

RET Record Element Type

SCM Source Code Management

SWT Standard Widget Toolkit

TDD Test Driven Development

UML Unified Modeling Language

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

Annotation

Annotationen dienen dazu, Metadaten innerhalb einer Programmiersprache oder DSL zu hinterlegen.

Artefakt

Ein Artefakt stellt im Rahmen dieser Arbeit ein oder mehre Quellcodedateien dar, die automatisiert erzeugt worden sind.

Artefakt-Generator

Ein Plug-in zur automatisierten Erstellung von Artefakten bzw. Quellcode. Der Generator nutzt dabei das Modell der DSL als Basis.

Domäne

Als Domäne wird der Bereich bezeichnet, in dem eine domänenspezifische Sprache eingesetzt wird.

Eclipse

Eine Entwicklungsumgebung für Java und andere Programmiersprachen

Function Point-Analyse

Methodik, die zur Aufwandsabschätzung von Softwareprojekten angewandt werden kann.

Lambda-Ausdruck

Ein Lambda-Ausdruck stellt eine anonyme Funktion dar, die an eine Methode übergeben werden kann.

Mockup

Beispielhafte Darstellung einer Anwendung ohne oder mit wenig Funktionalität.

Modell

Das Modell bildet mit Hilfe der DSL die Anforderungen einer Domäne in einer textuellen Form ab.

transient

Ein Element (Domäne, Attribut o.ä.), das zur Laufzeit nicht in einer Datenbank persistiert wird.

Literaturverzeichnis

[All12] OSGi Alliance. OSGi Alliance specification. Spezifikation für OSGi Release 5, 2012. URL: http://www.osgi.org/Specifications/HomePage.

[Art04] John Arthorne. Project builders and natures. 2004. URL: http://www.eclipse.org/articles/Article-Builders/builders.html.

[Bal09] Helmut Balzert. *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. 2. Auflage, 2009, Seiten 529–539.

[BCD+12] Ryan Bigg, Fredreick Cheung, Tore Darell, Jeff Dean, Mike Gunderloy, and Mikel Lindsaar. Getting started with rails. Getting Up and Running Quickly with Scaffolding, 2012. URL: http://guides.rubyonrails.org/getting_started.html#getting-up-and-running-quickly-with-scaffolding.

[Cer05] Gary Cernosek. A brief history of Eclipse. 2005. URL: http://www.ibm.com/developerworks/rational/library/nov05/cernosek/.

[Cle10] Torsten Cleff. *Basiswissen Testen von Software*. W3L-Verlag, 1. Auflage, 2010, Seiten 62–63.

[Cor09] Microsoft Corporation. The Microsoft code name M Modeling Language Specification - Introduction. 2009. URL: http://msdn.microsoft.com/en-us/library/dd285271.aspx.

[Cornta] Microsoft Corporation. Attributes (C# and Visual Basic). unbekannt. URL: http://msdn.microsoft.com/de-de/library/vstudio/z0w1kczw.aspx.

[Corntb] Microsoft Corporation. Quadrant Overview. unbekannt. URL: http://msdn.microsoft.com/en-us/library/dd857506(VS.85).aspx.

[Corntc] Microsoft Corporation. Visualization and Modeling SDK - Domain-Specific Languages. unbekannt. URL: http://msdn.microsoft.com/en-us/library/bb126259.aspx.

[Ecl07] Eclipse. The project description file. 2007. URL: http://help.eclipse.org/juno/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Freference%2Fmisc%2Fproject_description_file.html.

[ED96] Christof Ebert and Reiner Dumke. *Software-Metriken in der Praxis*. Springer Verlag, 1. Auflage, 1996, Seite 122.

[Eff10] Sven Efftinge. Xbase - A new programming language? 2010. URL: http://blog.efftinge.de/2010/09/xbase-new-programming-language.html.

[Eff12] Sven Efftinge. Xtend documentation. 2012. URL: http://www.eclipse.org/xtend/documentation.html.

[Eff13] Sven Efftinge. Xtext documentation. 2013. URL: http://www.eclipse.org/Xtext/documentation.html.

[HT06] Andrew Hunt and David Thomas. *The Pragmatic Programmer*. Addison-Wesley Professional, 19. Edition, 2006, Seite 103.

[iA08] itemis AG. Xtext.xtext. 2008. URL: https://github.com/eclipse/xtext/blob/master/plugins/org.eclipse.xtext/src/org/eclipse/xtext/Xtext.xtext.

[iA10a] itemis AG. Xbase.xtext. 2010. URL: https://github.com/eclipse/xtext/blob/master/plugins/org.eclipse.xtext.xbase/src/org/eclipse/xtext/xbase/Xbase.xtext.

[iA10b] itemis AG. Xtend.xtext. 2010. URL: https://github.com/eclipse/xtext/blob/master/plugins/org.eclipse.xtend.core/src/org/eclipse/xtend/core/Xtend.xtext.

[Mic07a] Sun Microsystems. Java 5 API. @ManyToMany Annotation, 2007. URL: http://docs.oracle.com/javaee/5/api/javax/persistence/ManyToMany.html.

[Mic07b] Sun Microsystems. Java 5 API. @ManyToOne Annotation, 2007. URL: http://docs.oracle.com/javaee/5/api/javax/persistence/OneToMany.html.

[Moi12] Kim Moir. The Architecture of Open Source Applications - Eclipse. 2012. URL: http://www.aosabook.org/en/eclipse.html.

[Ora10a] Oracle. Annotations. 2010. URL: http://docs.oracle.com/javase/1.5.0/docs/guide/language/annotations.html.

[Ora10b] Oracle. Core j2ee patterns - data access object. 2010. URL: http://www.oracle.com/technetwork/java/dataaccessobject-138824.html.

[Vö10] Markus Völter. Modellgetriebene, Komponentbasierte Softwareentwicklung. *JavaMagazin*, 2010. Online unter http://www.voelter.de/data/articles/MDSDandCBD-Part1.pdf.