

Methodische Entwicklung von Modellierungswerkzeugen

Jens Gulden, Stefan Strecker

Vorwort der Workshopleitung	346
<i>Matthias Kunze, Mathias Weske</i>	
Entwurf domänenspezifischer Modelle im Web mit Oryx	350; 2954–65
<i>Tony Modica, Enrico Biermann, Claudia Ermel</i>	
An Eclipse Framework for Rapid Development of Rich-featured GEF Editors based on EMF Models	351; 2966–79
<i>Eugen Wachtel, Marco Kuhrmann, Georg Kalus</i>	
A Domain Specific Language for Project Execution Models	352; 2980–94
<i>Jens Gulden</i>	
Minimal-invasive generative Entwicklung von Modellierungswerkzeugen mit dem Eclipse Graphical Modeling Framework (GMF)	353; 2995–09

Vorwort – Methodische Entwicklung von Modellierungswerkzeugen

Jens Gulden, Stefan Strecker

Universität Duisburg-Essen, Wirtschaftsinformatik und Unternehmensmodellierung
Universitätsstr. 9, 45141 Essen
{jens.gulden,stefan.strecker}@uni-duisburg-essen.de

In vielen Bereichen des Lebens und wissenschaftlicher Forschung spielt der Begriff „Modell“ eine zentrale Rolle. Modelle dienen zur Erfassung und Mitteilung spezifischen Wissens und können in unterschiedlichster Ausprägung vorliegen. Entsprechend vielfältig ist die Verwendungsweise des Begriffs und die darunter verstandenen Gegenstände. Klassischerweise wird die Herkunft des Begriffs aus dem lateinischen „modulus“ und der Tradition der Gebäudearchitektur abgeleitet, in der die Verwendung des Worts „Modello“ bis weit in das Mittelalter zurückverfolgt werden kann.¹

Kinderspielzeuge als Verkleinerungen von Kulturgegenständen sind seit Anbeginn der Menschheit üblich, ebenso wie Architekturmodelle und andere Nachbildungen realer Gegenstände. Neben einer physisch verkleinerten Abbildung bezeichnet der Begriff „Modell“ aber auch begriffliche Konstrukte. In der ersten Hälfte des 20. Jahrhunderts flossen Abstraktionen des physischen Modellbegriffs in die Natur- und Geisteswissenschaften ein, seitdem gehört beispielsweise die Rede von Atommodellen, Evolutionsmodellen, Sozialisationsmodellen oder semiotischen Modellen zur methodischen Terminologie unterschiedlicher wissenschaftlicher Fachbereiche.

Für Wissenschaften wie die Informatik oder Wirtschaftsinformatik spielen Modelle als Mittel zur Erfassung und Mitteilung von Wissen in doppelter Hinsicht eine wesentliche Rolle. Zum einen werden Modelle zur Erfüllung des Kernbereichs Informatik-bezogener Aufgaben methodisch eingesetzt, um auf Instanz-Ebene formale Beschreibungen von Diskursbereichen auszudrücken, die als Grundlage der Entwicklung von Softwaresystemen dienen können. Die zweite Seite wissenschaftlichen Interesses an Modellen besteht in der Reflexion über verwendete Modellierungstechniken und den Entwurf von Verfahren, um Modellierung effizienter zu gestalten und neue semantische Ausdrucksmöglichkeiten beim Modellieren zu entfalten. Die Einnahme dieser methodologischen Perspektive stellt eine genuine Aufgabe der Informatik und Wirtschaftsinformatik dar, und die theoretische Reflexion über Modellierungsmethoden ist eines ihrer zentralen Forschungsfelder.²

¹Vgl. Horst Bredekamp. Modelle der Kunst und der Evolution. In: Debatte 2 – Modelle des Denkens: Streitgespräch in der Wissenschaftlichen Sitzung der Versammlung der Berlin-Brandenburgischen Akademie der Wissenschaften am 12. Dezember 2003, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, 2005, S. 14

²Vgl. Wolfgang Hesse, Heinrich C. Mayr: Modellierung in der Softwaretechnik: eine Bestandsaufnahme. In: Informatik Spektrum 31(5): 377–393, 2008, S. 378: „Die Informatik kann man als die Disziplin der Modellbildung schlechthin bezeichnen [...]“

Jede Forschungsaktivität in diesem Bereich muss aber die grundsätzliche Annahme treffen, dass die Anwendung einer Modellierungssprache überhaupt effizient möglich ist. Die Frage der effizienten Anwendbarkeit einer Modellierungssprache stellt sich dabei nicht nur vor einem kognitiv-psychologischen Hintergrund im Sinn der Verständlichkeit und intellektuellen Handhabbarkeit der Sprache. Sinnlogisch vorausgesetzt ist ebenfalls die Verfügbarkeit eines Modellierungswerkzeugs, ohne das eine Modellierungssprache nicht nutzbringend verwendet kann. Auf dem Workshop „Methodische Entwicklung von Modellierungswerkzeugen“ widmen sich vier Beiträge diesem Aufgabenfeld und beleuchten unterschiedliche Aspekte, die bei der Erstellung von Modellierungswerkzeugen relevant sind.

Der Beitrag „Entwurf domänenspezifischer Modelle im Web mit Oryx“ stellt den Web-basierten Ansatz eines Modellierungs- und Metamodellierungswerkzeugs vor, der mit einer innovativen technologischen Umsetzung besticht. Das beschriebene Werkzeug öffnet den Blick auf mögliche zukünftige Entwicklungen und macht deutlich, dass die gegenwärtig verbreiteten Ansätze zur Erstellung von Modellierungswerkzeugen noch keinesfalls einen Endpunkt der Entwicklung darstellen.

„An Eclipse-Framework for Rapid Development of Rich-featured GEF Editors based on EMF Models“ stellt eine Methode zur Erzeugung grafischer Editoren im Rahmen des Eclipse Modeling Framework (EMF) und Eclipse Graphical Editing Framework (GEF) vor, und bietet damit einen methodisch alternativen Ansatz zum weit verbreiteten Eclipse Graphical Modeling Framework (GMF). Die Arbeit operiert mit einem Mapping-Verfahren zwischen semantisch-konzeptuellem Modell und grafisch-symbolischer Notation, das wesentliche Erweiterungen in die Modell-Notation einführt.

Im Beitrag „A Domain Specific Language for Project Execution Models“ wird die Entwicklung einer domänenspezifischen Sprache wissenschaftlich präzise rekonstruiert und die Herausforderungen und Lösungsstrategien bei einer exemplarischen Sprachentwicklung methodisch aufbereitet. Damit vermittelt der Beitrag einen transparenten methodologischen Überblick über den Entwicklungsprozess einer domänenspezifischen Sprache.

Eine Detailbetrachtung liefert der Beitrag „Minimal-invasive generative Entwicklung von Modellierungswerkzeugen mit dem Eclipse Graphical Modeling Framework (GMF)“. Hier wird auf Aspekte manueller Änderungen an generativ erzeugtem Quellcode eingegangen und ein Verfahren vorgeschlagen, um die Transformation von Modellen zu Quellcode mit dem Anfertigen manueller Änderungen methodisch zu vereinen.

Wir danken allen Autorinnen und Autoren für die Einreichung ihrer Beiträge – auch denjenigen, die letztlich nicht angenommen wurden. Den Mitgliedern des Programmkomitees, Ulrich Frank, Holger Giese, Jürgen Jung, Steffen Kruse, Yu Li, Jens von Pilgrim, Torsten Schlichting, sowie allen Gutachtern, danken wir für die Übernahme der Referate. Wir freuen uns mit ihnen auf einen spannenden und zielführenden Workshop „Methodische Entwicklung von Modellierungswerkzeugen“.

Essen, 1. Juli 2009,

Jens Gulden, jens.gulden@uni-duisburg-essen.de

Stefan Strecker, stefan.strecker@uni-duisburg-essen.de

Preamble – Methodical Development of Modelling Tools

Jens Gulden, Stefan Strecker

Universität Duisburg-Essen, Wirtschaftsinformatik und Unternehmensmodellierung
Universitätsstr. 9, 45141 Essen, Germany
{jens.gulden,stefan.strecker}@uni-duisburg-essen.de

The concept of “model” plays a key role in many areas of life and scientific research. Models capture and convey a specific kind of knowledge and can be represented in a variety of occurrences. Correspondingly, the term is used in different ways, applied to different purposes and meant to denote different subject matter. Historically, the origin of the concept is derived from the Latin “modulus” and the traditions of architecture, where the word “Modello” can be traced back to early medieval times.

Children’s toys as miniatures of cultural objects are common since the beginning of mankind, just like architecture models and other reproductions of real objects. Besides physically scaled-down replica, the idea of a “model” can also denote conceptual artefacts. In the first half of the 20th century, abstractions of the concept of a physical model entered natural sciences and humanities. Since then we refer to, e.g., the atomic model, the DNA model, models of human evolution, socialisation models or semiotic models in various scientific disciplines.

For information sciences such as Computer Science, Informatics or Business & Information Systems Engineering, models are of importance in two respects: On the one hand, models are a key instrument applied to represent formal descriptions of software systems at the instance level. On the other hand, scientific interest on models pertains to the reflection about modelling techniques and the design of modelling methods to make the act of modelling more efficient and to add to the semantic expressiveness of modelling languages. This rather methodological perspective represents a genuine task of the information sciences, and theoretic reflection about modelling methods is considered one of their main fields of research.

However, each research activity in this field must make a fundamental assumption representing a necessary precondition to scientific modelling research and to practical modelling projects: Namely that a modelling language can be efficiently applied. The question of efficient applicability of a modelling language not only refers to the cognitive, psychological aspects in the sense of comprehensibility and intellectual manageability. It also presupposes the availability of a (computer-based) modelling tool. Without such a tool, a modelling language cannot be beneficially put to use. The proceedings of the workshop “Methodical Development of Modelling Tools” comprises four papers devoted to this field

of research. The four contributions highlight different aspects relevant to developing modelling tools.

The paper “Entwurf domänenspezifischer Modelle im Web mit Oryx” (Design of domain-specific models on the Web with Oryx) discusses a Web-based approach characterised by its innovative technological realisation inside a Web browser. The tool directs the view towards potential future developments in modelling research and reveals that present approaches are not the end of technological developments.

“An Eclipse-Framework for Rapid Development of Rich-Featured GEF Editors based on EMF Models” presents a method for generating graphical editors in the context of the Eclipse Modeling Framework (EMF) and the Eclipse Graphical Editing Framework (GEF), and thus offers an alternative methodical approach to the widespread Graphical Modeling Framework (GMF). The presented research operates with a mapping-technique between the semantic conceptual model and graphical symbolic notation, introducing essential enhancements into model notation.

In the article “A Domain Specific Language for Project Execution Models”, the development process of a domain specific modelling language is precisely reconstructed, and challenges and solutions arising during the exemplified language development are methodically presented. This way, the article conveys a traceable methodological overview on the development process of a domain specific modelling language.

“Minimal-invasive generative Entwicklung von Modellierungswerkzeugen mit dem Eclipse Graphical Modeling Framework (GMF)” (Minimal-invasive generative development of modelling tools with the Eclipse GMF) investigates the problem of applying manual changes to source code automatically generated by a modelling tool. The paper suggests a procedure which methodically unifies the transformation of models to source code considering the creation of manual modifications.

We would like to express our thanks to all authors for their submissions – including those whose papers were eventually not accepted for publication. We are grateful to the members of the programme committee, Ulrich Frank, Holger Giese, Jürgen Jung, Steffen Kruse, Yu Li, Jens von Pilgrim, Torsten Schlichting, as well as to the referees, for dedicating their time and effort to review the submissions and are looking forward to an interesting and inspiring workshop on “Methodical Development of Modelling Tools”.

Essen, 1st July 2009,

Jens Gulden, jens.gulden@uni-duisburg-essen.de

Stefan Strecker, stefan.strecker@uni-duisburg-essen.de

Entwurf domänenspezifischer Modelle im Web mit Oryx

Matthias Kunze, Mathias Weske

Hasso Plattner Institut an der Universität Potsdam
Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam
{matthias.kunze,mathias.weske}@hpi.uni-potsdam.de

Die stetige Zunahme der Komplexität von Informationssystemen erschwert den Systementwurf mit herkömmlichen Methoden. Universelle Modellierungssprachen wie UML stoßen an ihre Grenzen, da sich die fachlichen Anforderungen und Eigenschaften eines derart entworfenen Systems oft in der Menge technischer Details verlieren.

Der Einsatz domänenspezifischer Modellierungssprachen verspricht Abhilfe: Der Systementwurf wird auf fachliche Konzepte statt seiner Implementierung abstrahiert und Domänenexperten werden befähigt, aktiv an Entwurf und Dokumentation des Systems teilzunehmen. Um domänenspezifische Modellierungssprachen effektiv einsetzen zu können, benötigt man jedoch Werkzeuge mit denen sich entsprechende Modelle entwickeln lassen. In dem Artikel wird die erweiterbare Modellierungsumgebung Oryx vorgestellt, die eine Vielzahl von Modellierungssprachen unterstützt, sowie die Umsetzung domänenspezifischer Modellierungssprachen in Oryx.

Eine Modellierungssprache besteht aus den Konzepten Notation, Syntax und Semantik [JS06]. Üblicherweise werden Syntax und Semantik durch formale bzw. semiformale Festlegungen definiert, z.B. durch das UML-Metamodell. Im Oryx Editor existieren *Stencil Sets* – textuell definierte Beschreibungen der graphischen Repräsentation der spezifischen Konzepte der Domäne (Notation) und Kompositionsregeln (Syntax) von Modellelementtypen. Semantik wird als Abbildung der Domänenkonzepte im Modell auf eine bestimmte Bedeutung verstanden. Dies wird in Oryx durch *Plugins* realisiert, mit denen sich Modelle beispielsweise abwickeln oder, im Rahmen modellgetriebener Ansätze, in andere Sprachen transformieren lassen.

In Oryx sind sowohl Anwendung als auch Modelle Ressourcen im Web. Das vereinfacht Installation und Datenhaltung und öffnet die Modellierungsplattform einem breiten Benutzerspektrum. Oryx besteht aus zwei Komponenten: Der *Editor* dient zur Entwicklung von Modellen verschiedener Modellierungssprachen. Mithilfe des *Repositorys* können entwickelte Modelle verwaltet, markiert (tagging) und bewertet (rating) werden. Auf dessen Grundlage wird auch die Zusammenarbeit mit Dritten (sharing) realisiert, indem Benutzern Rechte zur Betrachtung oder Bearbeitung von Modellen eingeräumt werden.

[JS06] Ethan K. Jackson und Janos Sztipanovits. Towards a formal foundation for domain specific modeling languages. In *EMSOFT '06: Proceedings of the 6th ACM & IEEE International conference on Embedded software*, Seiten 53–62, New York, NY, USA, 2006. ACM.

An ECLIPSE Framework for Rapid Development of Rich-featured GEF Editors based on EMF Models

Tony Modica, Enrico Biermann, Claudia Ermel
Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin
E-Mail: {modica,enrico,lieske}@cs.tu-berlin.de

Model-based development has an increasing importance in modern software engineering and other domains. Visual models such as Petri nets and UML diagrams proved to be an adequate way to illustrate many structural and behavioral system properties. However, while tooling for textual modeling is pretty mature now, visual tool builders are faced with a much higher complexity regarding the representation of model properties, and the interplay of the concrete syntax (the views) with the underlying abstract model representation, e.g. based on Java, XML or the Eclipse Modeling Framework (EMF). In order to ease the development of visual editors, the Graphical Editing Framework (GEF) provides the layout and rendering toolkit DRAW2D for graphics and follows the model-view-controller (MVC) architectural pattern to synchronize model changes with its views and vice versa. GEF's architecture allows to integrate models based on EMF, Java or XMI with their visual views and editors.

Generally, with a complex framework as GEF there are always many different possibilities to approach the implementation of a feature. Several frameworks support generation of code for from abstract editor specifications, like the ECLIPSE Graphical Modeling Framework (GMF) or MOFLON. Unfortunately, the visual languages we usually want to implement editors for and their simulation operations seem not appropriate to be specified in these frameworks, which assume models to be displayed in a single pane only. Moreover, if we used e.g. GMF to generate code as far as possible, GEF apprentices without deeper knowledge of the mechanisms in GEF would surely struggle when laying hand on the generated code to extend it with complex features.

Our research group has its main focus on applying formal techniques to visual modeling languages. We used our past experiences in teaching the development of GEF-based editors to generalize recurring code fragments for many editor features into code templates and to document them properly for simplifying the familiarization process for the students as well as the editor implementation. This development lead to our GEF-based framework MUVITORKIT (Multi-View Editor Kit). MUVITORKIT supports nested models, models needing multiple graphical viewers, and animated simulation of model behavior. Its architecture is designed in a way that encapsulates complex underlying mechanisms in GEF and simplifies the interaction with the ECLIPSE workbench.

A Domain Specific Language for Project Execution Models

Eugen Wachtel, Marco Kuhrmann, Georg Kalus

Technische Universität München, Institut für Informatik – I4
Boltzmannstr. 3, 85748 Garching, Germany
{wachtel|kuhrmann|kalus}@in.tum.de

The modeling of strategies for deriving valid project plans is a core task in development process's design. Strategies are used for planning and for the validation of concrete project plan-instances. Especially if using delivery-oriented planning, project managers need to know if they have all required deliverables available or if a scheduled project satisfies the process's requirements.

In this paper, we present a domain specific language (DSL) to model and validate a project plan as well as its execution, which is specified in form of a template called process execution strategy. The modeling of strategies is usually done by process engineers who require a deep understanding of the required workflows as well as knowledge about the underlying meta-model and the allowed processes. Especially the second part can be enforced or at least validated by specific software, making the modeling of complex strategies more manageable. Our DSL supports process engineers during the process design, in process improvement or introduction projects, allowing them to define and modify strategies and by providing real-time validation feedback visualizing inconsistencies.

To give a relevant example, we introduce a Microsoft DSL-Tools-based language for the V-Modell XT 1.3 describing its project execution strategies (PES). The meta-model for PES in the V-Modell XT is hierarchical and highly modular, which makes the modeling of strategies for the process engineer rather complicated and fault-prone. Therefore, the V-Modell XT is a good example for our DSL.

To introduce the implementation of our prototype we present a domain model with its two facets for modeling project execution strategies and project plans. As the strategy definitions part is based on the V-Modell XT meta-model, we investigate and propose the required extensions and/or restrictions needed to guarantee a valid modeling of strategies. Further, we explain how the domain model can be transformed into the Microsoft DSL-Tools domain model. We also present the DSL-Tools designer and highlight a couple of noteworthy implementation details.

Minimal-invasive generative Entwicklung von Modellierungswerkzeugen mit dem Eclipse Graphical Modeling Framework (GMF)

Jens Gulden

Universität Duisburg-Essen, Wirtschaftsinformatik und Unternehmensmodellierung
Universitätsstr. 9, 45141 Essen
jens.gulden@uni-duisburg-essen.de

Generative Software-Entwicklung aus Modellen und die anschließende manuelle Bearbeitung von Quellcode erscheinen oft als methodisch gegenläufige Verfahren. Die vorliegende Arbeit stellt einen Ansatz zur generativen Entwicklung graphischer Modellierungswerkzeuge mit dem Eclipse Graphical Modeling Framework (GMF) vor, der automatische Code-Generierung und manuelle Programmierung methodisch vereint und im Rahmen eines gemeinsamen Vorgehensmodells beide als parallele konstitutive Bestandteile eines Software-Entwicklungsprozesses auffasst.

Die Vereinbarkeit von automatischer Code-Generierung und manuellen Modifikationen am generierten Quellcode wird erreicht, indem die ursprüngliche Problemstellung von Fragen der Synchronisation zwischen Ausgangsmodell und Quellcode auf die Explizierung manueller Änderungsschritte verlagert wird. Änderungen am Quellcode werden in einem übergreifenden Modifikationsmodell erfasst, so dass sie unabhängig vom Code-Generierungsprozess formal beschrieben sind und zu jedem Zeitpunkt reproduzierbar bleiben. Dieses methodische Vorgehen vermeidet zyklische Abhängigkeiten im Informationsfluss des Entwicklungsprozesses. Revisionen der Ergebnisse früherer Entwicklungsphasen, insbesondere lokale Änderungen am Metamodell und an Code-Generierungstemplates, bleiben möglich.

Eine prototypische Umsetzung des Verfahrens wird zum Abschluss des Artikels vorgestellt.