

Meta-Modellierung im praktischen Einsatz

Jens Weller, Kai K. Großmann, Knut Großmann

Technische Universität Dresden
Institut für Werkzeugmaschinen und Steuerungstechnik
01062 Dresden

{weller|kkg|groszman}@iwm.mw.tu-dresden.de

Abstract: Auf dem Markt der Meta-Modellierungswerkzeuge stehen sich generische Werkzeuge, Generatoren und Frameworks gegenüber. Die Abschätzung von Aufwand und Nutzen der unterschiedlichen Ansätze fällt jedoch aufgrund fehlender Erfahrungsberichte schwer. Um dieses Defizit zu beseitigen, stellt der folgende Beitrag ein generisches Werkzeug vor und beschreibt die Sprachanpassung sowie die Modellierung mit dem Werkzeug im praktischen Einsatz.

1 Einleitung

Bereits seit vielen Jahren sind generische Modellierungswerkzeuge auf dem Markt, welche die Definition und Anpassung der Modellierungssprache erlauben. Hier wird im Werkzeug selbst zunächst die Sprache beschrieben. Anschließend steht das angepasste Modellierungswerkzeug entweder sofort zur Verfügung oder wird auf Knopfdruck erstellt [Fi05, Cu10]. In den letzten Jahren etablieren sich darüber hinaus verstärkt Frameworks, wie EMF [Ec10], welche die Generierung von Modellierungswerkzeugen unterstützen und die automatisierte Weiterverarbeitung der Modelle vereinfachen.

Gerade in Praxisprojekten, die im Gegensatz zur Wissenschaft, nicht primär die Modellierung fokussieren, sondern in denen Modelle lediglich Hilfsmittel darstellen, stellt die schnelle Umsetzung und Anpassung des Modellierungswerkzeugs einen wesentlichen Erfolgsfaktor dar. Aufgrund fehlender Erfahrungsberichte in der Literatur kann der Anwender jedoch nur schwer Aufwand und Nutzen der oben erwähnten Ansätze zur Werkzeuggenerierung abschätzen.

Um dieses Defizit zu beheben, wird im Folgenden der Einsatz des Cubetto Toolsets [Cu10], als Vertreter der generischen Meta-Modellierungswerkzeuge, in einem Forschungsprojekt des Maschinenbaus vorgestellt. Dabei werden die Kriterien für die Auswahl des Werkzeugs offengelegt (Abschnitt 2) und die Sprachanpassung, die Modellierung sowie die Modellnutzung beschrieben (Abschnitt 3). Die diskutierten Vor- und Nachteile dienen nicht nur der Entscheidungsunterstützung bei der Auswahl eines Ansatzes, sondern können auch für die zukünftige Entwicklung von Meta-Modellierungswerkzeugen bzw. entsprechenden Generatoren genutzt werden.

2 Werkzeugauswahl

2.1 Projektbeschreibung und Anforderungen

Im Rahmen der Forschungsarbeiten des DFG-Sonderforschungsbereichs 639 [Huf06], werden Methoden entwickelt, mit denen bei neuen, innovativen Prozessketten die Reproduzierbarkeit der Fertigung abgesichert werden soll. Um dies erreichen zu können, sollen Prozessmodelle erstellt und deren Parameter (anhand von Daten, die während exemplarischer Prozessdurchführungen gewonnen werden) fortlaufend präzisiert werden, bis ein „eingefahrener“ Zustand erreicht worden ist. Modellierung und Datenanalyse greifen hierbei ineinander und sollen dazu dienen, ergebnisrelevante Wechselwirkungen zwischen den einzelnen Prozessparametern identifizieren zu können. Dieses empirisch erlangte Erfahrungswissen wird anschließend, u. a. für die Simulation und (maschinelle) Steuerung der Prozesse genutzt [GW07].

Um diese Forschungsarbeiten zu unterstützen, soll eine geeignete Softwarelandschaft entwickelt werden. Für die Modellierung der Prozesse wurde der Einsatz eines Modellierungswerkzeugs beschlossen. Im Gegensatz zu „eingefahrenen“ Prozessketten, sind innovative Prozessketten jedoch dadurch gekennzeichnet, dass sie die Bedingungen hinsichtlich Durchgängigkeit, Definiertheit, Beherrschbarkeit, Analysierbarkeit und Steuerbarkeit zunächst nicht oder nicht vollständig erfüllen. Aus diesem Grund ist auch die Modellierungssprache, welche für die Dokumentation der Prozesse genutzt werden soll, zu Beginn des Projektes noch nicht abschließend definiert. Stattdessen entsteht sie „parallel“ zur Entwicklung der Prozessketten. Daher musste ein Modellierungswerkzeug gewählt werden, welches die *fortlaufende Anpassung der Modellierungssprache* erlaubt und bereits bestehende Prozessmodelle entsprechend nachführen kann.

Neben dem Modellierungswerkzeug kommen auch weitere Werkzeuge, u. a. für die Datengewinnung bei der exemplarischen Durchführung von Prozessen, zum Einsatz. Da sich diese Aktivitäten direkt auf die modellierten Prozessketten beziehen, ist eine Kommunikation des Modellierungswerkzeugs mit dem Datengewinnungswerkzeug notwendig. In diesem Zusammenhang und auch unter Berücksichtigung zukünftiger Einsatzfelder ist es notwendig, dass das Modellierungswerkzeug über *dokumentierte Schnittstellen* verfügt.

Die Gesamtprozesskette im Projekt wird von 17 Teilprojekten in 11 unterschiedlichen Struktureinheiten mit einer Vielzahl von Mitarbeitern erforscht [Huf06]. Die inhaltliche, organisatorische und räumliche Trennung der einzelnen Projektmitarbeiter erfordert ein *verteiltes Modellierungssystem* mit integriertem *Benutzermanagement* (Rollen- und Rechtssystem). Außerdem sollen über ein *Versionskontrollsystem* Arbeitsstände versioniert und auseinanderlaufende Arbeitsstände einzelner Benutzer mit der Hauptversion wieder zusammengeführt werden.

Letztendlich waren bei der Auswahl eines geeigneten Modellierungswerkzeugs auch die Anschaffungskosten von Bedeutung. Da im Rahmen des Forschungsprojektes keine Mittel für die Anschaffung von entsprechender Software vorgesehen waren, musste ein Werkzeug gefunden werden, für das *keine Lizenzgebühren* anfallen.

2.2 Auswahl

Im Rahmen des Projekts wurden verschiedene Modellierungswerkzeuge (aus der Wirtschaftsinformatik und der technischen Prozessgestaltung) gegen die o. g. Anforderungen geprüft, die jedoch die geforderten Anforderungen nicht vollständig erfüllen konnten. Die Alternative einer Eigenentwicklung (auf Basis bestehender Frameworks), wurde aufgrund des hohen Aufwands nicht weiter verfolgt. Stattdessen wurde einer „Konfigurationslösung“ auf Basis des generischen Modellierungswerkzeugs Cubetto Toolset [Cu10] der Vorzug gegeben, da dieses nicht nur sämtliche o. g. Anforderungen erfüllt, sondern sich der Anbieter auch in direkter räumlicher Nähe befindet, wodurch Supportanfragen unmittelbar beantwortet werden können. Das Cubetto Toolset ermöglicht die Erstellung individueller Modelleditoren. Dabei wird das Werkzeug lediglich konfiguriert, eine Programmierung ist nicht erforderlich. Dies hat den Vorteil, dass sowohl die Modellierungssprache als auch das zugehörige Modellierungswerkzeug vom Methodenentwickler angepasst werden können.

3 Der Einsatz des Modellierungswerkzeugs im Projekt

3.1 Methoden- und Werkzeugentwicklung

Das Cubetto Toolset basiert auf der E³-Methode, einer Methode zur Methodenentwicklung [Cu10]. Die Konfiguration des Werkzeugs orientiert sich daher an der E³-Notation, einer Sprache zur Beschreibung von Modellierungsmethoden [Gr04]. Diese Sprache beinhaltet Elemente für die Beschreibung von Konzepten, deren Eigenschaften, Beziehungen sowie die Definition von Sichten und der grafischen Darstellung der Konzepte.



Abbildung 1: Konfiguration des Cubetto Toolsets mit Hilfe grafischer Meta-Modelle

Für die Beschreibung der Modellierungssprache im Cubetto Toolset stehen verschiedene Editoren zur Verfügung. Die Sprache kann entweder über ein grafisches Meta-Modell beschrieben oder über eine baumartige Struktur konfiguriert werden. Im ersten Fall wird das Modellierungswerkzeug selbst verwendet, um das Meta-Modell zu erzeugen, als Sprache kommt die E³-Notation zum Einsatz (vgl. Abbildung 1).

Anschließend wird das Meta-Modell transformiert und die Sprache steht für die Modellierung im Werkzeug zur Verfügung. Bestehende Modelle werden dabei nicht automatisch in die neue Sprache übertragen. Im zweiten Fall wird das Werkzeug direkt modifiziert, was sich auch auf bestehende Modelle auswirkt. Im beschriebenen Projekt wurden beide Ansätze verwendet. Für die initiale Erstellung des Werkzeugs zu Beginn des Projekts wurde der grafische Ansatz gewählt, da die verwendeten Meta-Modelle die Diskussion über Konzepte (zu Beginn ca. 10) und deren Beziehungen vereinfachen. Bei Erweiterungen, die sich während der Anwendung der Methode ergaben, haben wir den zweiten Weg gewählt, da hierbei die bereits bestehenden Modelle weiter verwendet werden konnten. Beide Möglichkeiten sind im Cubetto Toolset sehr schnell umzusetzen. Sofern die sprachlichen Konzepte bekannt sind und lediglich die Werkzeugkonfiguration durchgeführt werden muss, ist die Hinterlegung von Sprachkonzepten und deren Beziehungen in wenigen Minuten realisiert. Etwas mehr Aufwand erfordert jedoch die Konfiguration der grafischen Darstellung der Konzepte.

Das Cubetto Toolset ist hier sehr mächtig und erlaubt komplexe Figuren, deren Verhalten und Darstellung sich in Abhängigkeit vom Modellinhalt ändern kann. So können Teilbereiche eines Modellelements ein- und ausgeschaltet werden oder Inhalte von verbundenen Modellelementen anzeigen. Dies erfordert jedoch einige Erfahrung. In unserem Fall hatte ein Cubetto-erfahrener Mitarbeiter die grafische Darstellung von ca. 10 Konzepten innerhalb von zwei Stunden umgesetzt, ein Neuling benötigt nach einem einführenden Workshop und der Möglichkeit persönlicher Rückfragen hierfür auch schon einmal zwei bis drei Tage. Insgesamt sind die Lernerfolge jedoch sehr hoch.

Neben der eigentlichen Sprachanpassung musste im beschriebenen Projekt auch die Funktion für den Export der Modelle erstellt werden. Hierfür stehen im Cubetto keine Konfigurationsmechanismen zur Verfügung. Stattdessen kommt ein Plugin-Konzept zum Einsatz. Dabei werden Java-Programme geschrieben, die auf die erstellte Modellierungssprache und die erstellten Modelle zugreifen können und diese modifizieren oder, wie in unserem Fall, in eine andere Anwendung exportieren können. Dies ist leider nur durch einen Programmierer möglich. Dieser wird allerdings mit einer API unterstützt, die sich direkt aus der erstellten Modellierungssprache ableitet. Damit ist der Zugriff auf die Modelle intuitiv, sofern die Sprache selbst bekannt ist.

3.2 Modellierung

Die Modellierung mit dem Cubetto Toolset orientiert sich am üblichen Anlegen und Verknüpfen von Modellelementen. Dabei sind die Modelle im Cubetto Toolset immer an eine konkrete Modellierungssprache gebunden. Entsprechend richten sich die verfügbaren Konzepte (vgl. Abbildung 2, rechte Seite) und möglichen Beziehungen nach den innerhalb der Sprache hinterlegten Regeln. Dabei können bereits während der Modellierung Syntaxprüfungen erfolgen, die z. B. das Verbinden von Elementen, welche keine Beziehung aufweisen dürfen, unterbinden. Darüber hinaus können, mit Hilfe von so genannten Triggern die bei der Sprachspezifikation definiert werden, Verknüpfungen von Modellelementen durch das Werkzeug auch automatisch angelegt oder entfernt werden, was die Bedienung durch den Modellierer wesentlich erleichtern kann.

So können bspw. beim Anlegen eines Diagramms, welches als Verfeinerung eines Prozesses dienen soll, die eingehenden und ausgehenden Objekte des Prozesses automatisch in der verfeinerten Darstellung erzeugt werden. Die verfügbaren Trigger und die damit verbundene (subjektive) Qualität der Modellierungsumgebung werden dabei maßgeblich von der hinterlegten Sprachspezifikation geprägt. Hat der Methodenentwickler hier mehr Aufwand investiert, sinkt folglich der Zeitbedarf bei der Modellierung.

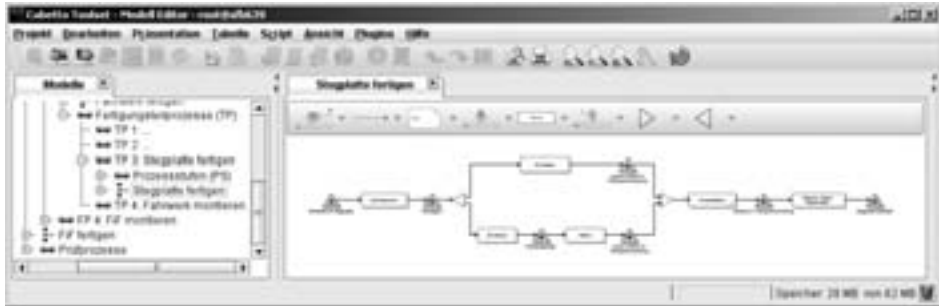


Abbildung 2: Nutzung der erstellten Modellierungssprache

Die Modellierung wird ergänzt um eine Druckfunktion, ein Benutzermanagement und ein Konfigurationsmanagement (KM). Insbesondere das KM wurde im vorliegenden Projekt zur verteilten Modellierung genutzt. Das Werkzeug besitzt eine Merge-Funktion zum Zusammenführen unterschiedlicher Arbeitsstände und zur Auflösung von Konflikten. Dies funktioniert sehr gut, allerdings erfordert insbesondere das Zusammenführen einige Erfahrung im Umgang mit dem System. Daher wird der Merge im Projekt nur zentral von einem Modellverwaltungsteam durchgeführt. Das vereinfacht den Ablauf, vermeidet unerwünschte Änderungen und entlastet die Modellierer in den Teilprojekten.

3.3 Methoden- und Werkzeugänderungen während der Modellierung

Um ungeplante Änderungen an der Modellierungssprache zu vermeiden, wurden zwei Mechanismen verwendet. Erstens wurde mit dem integrierten Benutzermanagement eine entsprechende Modifikation für Modellierer unterbunden. Zweitens wurde ein Lenkungsausschuss [Di04], bestehend aus erfahrenen Modellierern, eingerichtet, der Änderungswünsche diskutiert und diese durch den Methodenentwickler umsetzen lässt. Die Änderungswünsche werden dabei in einem Bugtracking System (Bugzilla [BU10]) gesammelt. Wie oben beschrieben, erfolgt die Anpassung des Werkzeugs direkt durch Modifikation im Werkzeug. Das hat den Vorteil, dass Änderungen an der Sprache direkt am bestehenden Modell überprüft und ausprobiert werden können. Dies bedeutet jedoch auch, dass beim Entfernen eines Konzepts entsprechende Modellelemente aus dem Modell gelöscht werden [EW07]. Wird ein Konzept im Zuge des Aufspaltens oder beim Zusammenlegen von Konzepten gelöscht, muss folglich eine entsprechend manuelle Nachbearbeitung der Modelle erfolgen. Alte Stände der Modelle (und damit auch der Sprache) stehen durch das integrierte KM-System jedoch jederzeit zur Verfügung, was eine solche Übertragung erleichtert.

4 Zusammenfassung

In diesem Beitrag wurde der Einsatz eines generischen Modellierungswerkzeugs in einem praktischen Projekt beschrieben. Diese Werkzeuggattung bietet den entscheidenden Vorteil, schnell konfigurierbar zu sein und damit für den praktischen Einsatz schnell zur Verfügung zu stehen. Darüber hinaus bringen generische Modellierungswerkzeuge häufig bereits umfangreiche Basisfunktionen, wie Druckfunktionen sowie Benutzer- und Konfigurationsmanagement mit. Dabei muss auf eine benötigte (automatisierte) Nutzung der Modelle nicht verzichtet werden. Allerdings ist bspw. beim Cubetto Toolsets hierfür eine Programmierung notwendig. Nachteile ergeben sich mitunter durch die Generik selbst, da die zahlreichen Möglichkeiten der Anpassung (z. B. Trigger) die Konfiguration des Werkzeugs für Anfänger erschweren und die Werkzeugqualität mindern können.

Zukünftige Werkzeugentwicklungen sollten sich diesem Problem stellen und komplexe Funktionen zu Bausteinen zusammenfassen. Auch eine integrierte Verwaltung von fertigen Methoden- und damit Werkzeugfragmenten könnte den Konfigurationsaufwand senken. Das im angewendeten Werkzeug integrierte Konfigurationsmanagement kann gerade für Meta-Modellierungswerkzeuge als essentiell angesehen werden. Auch hier kann eine auf den (Fach-)Modellierer zugeschnittene Benutzeroberfläche jedoch die Bedienung erleichtern.

Literaturverzeichnis

- [Bu10] Mozilla Foundation: Bugzilla, <http://www.bugzilla.org>. Download: 06.04.2010
- [Cu10] Cubetto Toolset, <http://www.semture.de/cubetto>. Download: 06.04.2010.
- [Di04] DIN Deutsches Institut für Normung e.V.: Qualitätsmanagement: Leitfaden für Konfigurationsmanagement (DIN ISO 10007:2003), Beuth Verlag, Berlin, 2004.
- [Ec10] Eclipse Modeling Framework Project, <http://www.eclipse.org/modeling/emf/>, Download: 06.04.2010.
- [EW07] Esswein, W., Weller, J.: Method Modifications in a Configuration Management Environment, In: (Österle, H., Schelp, J., Winter, R.): Proc. 15th European Conf. on Information Systems, St. Gallen, 2007, S. 2002-2013.
- [Fi05] Fill, H.-G.: UML Statechart Diagrams on the ADONIS Metamodeling Platform, In: Electronic Notes in Theoretical Computer Science, 127, 2005, S. 27-36.
- [Gr04] Greiffenberg, S.: Methodenentwicklung in Wirtschaft und Verwaltung, Verlag Dr. Kovac, 2004.
- [GW07] Großmann, K.; Wiemer, H.: Modellierung der Prozesskette für textilverstärkte Verbund-Bauteile: Prozesscharakter, Modellanforderungen und Beschreibungsmittel, In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, 03, 2007.
- [HMO94] Harmsen, F.; Brinkkemper, S.; Oei, J. L. H.: Situational method engineering for information system project approaches. In (Olle, T. W., Hrsg.): Methods and associated tools for the information systems life cycle, 1994, S. 169-194.
- [Huf06] Hufenbach, W.: Textilverstärkte Verbundkomponenten für funktionsintegrierende Mischbauweisen bei komplexen Leichtbauanwendungen, Sonderforschungsbereich 639 der DFG an der TU Dresden, 10. Dresdner Leichtbausymposium, Dresden, 2006.
- [RRD04] Ralyté, J.; Rolland, C.; Deneckère, R.: Towards a meta-tool for change-centric method engineering: A typology of generic operators, Lecture Notes in Computer Science 3084, 2004, S. 202-218.