

# Entwicklung einer objektorientierten Prozessmodellierungssprache

Vom Fachbereich Maschinenbau  
an der Technischen Universität Darmstadt  
zur  
Erlangung des Grades eines Doktor–Ingenieurs (Dr.–Ing.)  
genehmigte

D I S S E R T A T I O N

vorgelegt von

**Dipl.–Ing. Jochen Raßler**

aus Leimen

Berichterstatter:	Prof. Dr.–Ing. Reiner Anderl
Mitberichterstatter:	Prof. Dr.–Ing. Eberhard Abele
Tag der Einreichung:	30. Juli 2015
Tag der mündlichen Prüfung:	8. Dezember 2015

Darmstadt 2015

D 17



Forschungsberichte aus dem Fachgebiet  
Datenverarbeitung in der Konstruktion

Band 54

**Jochen Raßler**

**Entwicklung einer objektorientierten  
Prozessmodellierungssprache**

D 17 (Diss. TU Darmstadt)

Shaker Verlag  
Aachen 2016

**Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Darmstadt, Techn. Univ., Diss., 2015

Copyright Shaker Verlag 2016

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-4200-9

ISSN 1435-1129

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

### **Ehrenwörtliche Erklärung**

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig angefertigt habe. Sämtliche aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Dissertation wurde bisher keiner anderen Prüfungsbehörde vorgelegt und noch nicht veröffentlicht.

Darmstadt, den 30. Juli 2015



---

# Vorwort des Herausgebers

Die moderne Informations- und Kommunikationstechnologie (IKT) bietet vielfältige Innovations- und Leistungspotenziale, die im Entstehungsprozess neuer Produkte auszuschöpfen sind. Dies setzt jedoch voraus, dass die wissenschaftlichen Grundlagen zum Einsatz der modernen Informations- und Kommunikationstechnologie in der Produktentstehung vorliegen und neue Methoden wissenschaftlich abgesichert sind. Darüber hinaus stellen die wissenschaftliche Durchdringung und die Bereitstellung wissenschaftlicher Forschungsergebnisse eine abgestimmte Kooperation zwischen Forschung und Industrie dar.

Vor diesem Hintergrund informiert diese Schriftreihe über aktuelle Forschungsergebnisse des Fachgebiets Datenverarbeitung in der Konstruktion (DiK) des Fachbereichs Maschinenbau an der Technischen Universität Darmstadt. Ziel der Forschungsarbeiten ist die wissenschaftliche Durchdringung innovativer, interdisziplinärer und integrierter Produktentstehungsprozesse und darauf aufbauend die Konzeption neuer Methoden für die Entwicklung, Konstruktion, Arbeitsvorbereitung und Herstellung neuer, innovativer Produkte.

Eine wesentliche Voraussetzung für medienbruchfreie Durchgängigkeit digitaler Produktdaten ist in diesem Zusammenhang die genaue Kenntnis der in den Phasen der Produktentstehung beteiligten Prozesse. Diese Prozesse müssen ebenso, wie die Produktdaten formal spezifiziert werden, was bedeutet, dass durch die formale Spezifikation dann ein Prozessdatenmodell entsteht, das automatisiert in geeignete Softwaresysteme zur Prozesssteuerung umgesetzt werden kann. Während zur formalen Spezifikation von Produktdatenmodellen bereits formale Spezifikationssprachen existieren, so ist dies für die Beschreibung von Prozessen für die Phasen der Produktentstehung nicht der Fall.

Diese Herausforderung greift Herr Dipl.-Ing. Jochen Raßler auf und entwickelt mit seiner Dissertation „Entwicklung einer objektorientierten Prozessmodellierungssprache“ einen Ansatz für eine objektorientierte Prozessmodellierungssprache zur formalen Spezifikation von Prozessen für die Produktentstehungsphasen. Seine

Dissertation legt damit wissenschaftliche Grundlagen, um innovative Produktentstehungsprozesse zu unterstützen.

Darmstadt, 2015

Reiner Anderl

---

# Vorwort des Autors

Prozesse nehmen in Unternehmen eine entscheidende Rolle ein. Sie definieren den Ablauf der Produktentstehung und koordinieren die Schnittstellen über die Phasen des Produktlebenszyklus hinweg. Klar definierte Prozesse sind dadurch der Garant für reibungsfreie Kommunikation und Reproduzierbarkeit der Abläufe.

Viele Unternehmen fordern klar definierte Prozesse, versuchen diese immer weiter zu optimieren und setzen verschiedene Workflow-Management-Tools ein, um auf Taskebene den Workflow zu beschreiben. Jedoch sind Workflow-Management-Tools einerseits zur Definition von Prozessen ungeeignet und andererseits stellen sie eine Anbieter-abhängige Sichtweise auf den Workflow dar, wobei diese Sichtweise mitunter eine Nachvollziehbarkeit der Prozesse nicht gewährleistet, da nur auf Task- oder Modulebene die Workflow-Beschreibung vorliegt. Darüber hinaus finden sich oft Prozessbeschreibungen in Microsoft Word, Excel oder PowerPoint, die als Dokumentation und Erklärung der Prozesse zugleich dienen. Diese Dokumentationen müssen aber manuell gepflegt und erstellt werden und haben keinerlei systemtechnische Kopplung mit den Workflowtools, sodass Abweichungen zwischen realisiertem und dokumentiertem Prozess durchaus die Regel sind. Nur selten werden aktuelle Prozessbeschreibungen in Prozessmodellierungssprachen wie EPK oder BPMN erstellt. Diese werden in der Regel im Nachhinein erstellt und oft auch nur, wenn gerade Ressourcen frei sind. Da sich aber die Prozesse und die Versionen der Workflow-Management-Tools meist schneller entwickeln, als Ressourcen für die Beschreibungen der Prozesse in den Prozessmodellierungssprachen bereit gestellt werden, liegen meist veraltete Prozessdefinition in spezialisierten Sprachen vor.

Diese Lücke versucht die vorliegende Dissertation zu adressieren, indem eine moderne Prozessmodellierungssprache entwickelt wird, die sowohl als Definitionssprache als auch als Basis zur Ableitung von Abläufen in Workflow-Management-Tools sowie zur finalen Dokumentation der Prozesse genutzt werden kann. Dabei wird ein Paradigmenwechsel in der Prozessmodellierung vorgeschlagen, den wir im Rahmen der Softwareentwicklung bereits vor über drei Jahrzehnten erleben durften. Dabei werden auf die Erfahrung und das technische Wissen aus dem Bereich der

Softwaremodellierung zurückgegriffen und etablierte Technologien als Basis für die Entwicklung der objektorientierten Prozessmodellierungssprache genutzt.

Leimen, 2015

Jochen Raßler

---

# Inhaltsverzeichnis

<b>Vorwort des Herausgebers</b>	<b>iii</b>
<b>Vorwort des Autors</b>	<b>v</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Ziele . . . . .	1
1.2 Aufbau der Dissertation . . . . .	4
<b>2 Stand der Technik</b>	<b>7</b>
2.1 Grundlagen . . . . .	7
2.1.1 Modelltheorie . . . . .	7
2.1.2 Prozess und Workflow . . . . .	11
2.1.3 Objektorientierung . . . . .	11
2.2 Modelle und Metamodelle . . . . .	13
2.2.1 UML und MOF . . . . .	13
2.2.2 Zustandsautomaten . . . . .	16
2.2.3 Petri-Netze . . . . .	17
2.3 Prozessmodellierung . . . . .	17
2.3.1 Modellierungsmethoden . . . . .	17
2.3.2 Modellierungssprachen . . . . .	19
2.3.2.1 IDEF . . . . .	19
2.3.2.2 EPK / eEPK . . . . .	20
2.3.2.3 Prozessmodellierung mit UML . . . . .	21
2.3.2.4 BPMN . . . . .	23
2.3.2.5 IUM . . . . .	23
2.3.2.6 PSL . . . . .	24
2.3.2.7 SOM . . . . .	25
2.3.2.8 Objekt-Z . . . . .	25
2.3.2.9 Forschungsansätze . . . . .	25
2.4 Fazit . . . . .	29

<b>3</b>	<b>Anforderungen an eine objektorientierte Prozessmodellierungssprache</b>	<b>33</b>
3.1	Entwicklungsziele . . . . .	33
3.2	Konkretisierung der Anforderungen . . . . .	34
3.2.1	Generelle Anforderungen . . . . .	34
3.2.2	Anforderungen aus der Objektorientierung . . . . .	35
3.2.3	Anforderungen an die Sprache und Notation . . . . .	36
3.3	Fazit . . . . .	38
<b>4</b>	<b>Konzeptionierung einer objektorientierten Prozessmodellierungssprache</b>	<b>41</b>
4.1	Herleitung einer objektorientierten Prozessmodellierungssprache . .	41
4.2	Designansätze für PML . . . . .	45
4.3	Diagramme und Elemente . . . . .	46
4.3.1	Assoziationen — Verknüpfungen zwischen Elementen . . . .	47
4.3.2	Ereignisse in PML . . . . .	50
4.3.3	Das Klassendiagramm der PML . . . . .	53
4.3.3.1	Allgemeines Design der PML-Klasse . . . . .	53
4.3.3.2	Konkrete Modellierungsmöglichkeiten mit der PML-Klasse . . . . .	56
4.3.4	Das Objekt- / Projektdiagramm . . . . .	59
4.3.5	Das Paketdiagramm . . . . .	60
4.3.6	Das Zustandsdiagramm . . . . .	61
4.3.7	Das Aktivitätsdiagramm . . . . .	63
4.4	Objektorientierte Prozessmodellierung mit PML . . . . .	66
4.4.1	Allgemeine Modellierungsmethodik . . . . .	66
4.4.2	Praxisrelevante Modellierungsprobleme . . . . .	66
4.4.2.1	Determinismus, Konvergenz und Projektmanagement	66
4.4.2.2	Deadlocks und Race Conditions . . . . .	67
4.4.2.3	Nebenläufigkeit, Synchronität und Asynchronität .	68
4.4.2.4	Das Diamantproblem . . . . .	69
4.4.2.5	Ausnahmebehandlung . . . . .	71
4.4.2.6	Berechenbarkeit . . . . .	71
4.4.3	Modell und Sichten . . . . .	72
4.4.4	Begriffliche Anmerkungen . . . . .	72
4.4.4.1	Prozess, Projekt und Aktivität . . . . .	72
4.4.4.2	Modellierungsparadigma . . . . .	73
4.4.4.3	PML als Sprache oder Methode? . . . . .	74
4.5	Fazit . . . . .	74
<b>5</b>	<b>Implementierung der PML</b>	<b>77</b>
5.1	Wahl des Entwicklungsansatzes . . . . .	77

5.2	PML als Instanz der CMOF . . . . .	78
5.3	Das PML-Metamodell . . . . .	79
5.3.1	Zusammenhang zwischen UML- und PML-Klassen . . . . .	80
5.3.2	Das Event . . . . .	81
5.3.3	Die Aktivität . . . . .	83
5.3.4	Die Ressource . . . . .	87
5.3.5	Die Prozess-Klasse . . . . .	88
5.3.6	Weitere PML-Klassen . . . . .	89
5.4	Fazit . . . . .	90
<b>6</b>	<b>Validierung und Verifikation der objektorientierten Prozessmodellierungssprache PML</b>	<b>91</b>
6.1	Validierung der PML . . . . .	91
6.1.1	Stücklisten-basierte Prozessmodellierung . . . . .	92
6.1.2	CiP — Die Prozesslernfabrik . . . . .	95
6.2	Verifikation der Anforderungen . . . . .	103
6.3	Fazit . . . . .	107
<b>7</b>	<b>Ausblick</b>	<b>109</b>
<b>8</b>	<b>Zusammenfassung der Dissertation</b>	<b>113</b>
	<b>Literaturverzeichnis</b>	<b>115</b>
<b>A</b>	<b>Die PML Spezifikation</b>	<b>129</b>
A.1	UML Infrastructure Integration . . . . .	129
A.2	Der PML Core . . . . .	140
A.2.1	Association . . . . .	143
A.2.2	BehavioralFeature . . . . .	156
A.2.3	Classifier . . . . .	158
A.2.4	Comment . . . . .	163
A.2.5	Constraint . . . . .	166
A.2.6	DirectedRelationship . . . . .	170
A.2.7	Element . . . . .	172
A.2.8	ElementImport . . . . .	175
A.2.9	Expression . . . . .	180
A.2.10	Feature . . . . .	182
A.2.11	Generalization . . . . .	184
A.2.12	InstanceSpecification . . . . .	187
A.2.13	InstanceValue . . . . .	192
A.2.14	LiteralBoolean . . . . .	194

A.2.15	LiteralInteger	196
A.2.16	LiteralNull	198
A.2.17	LiteralSpecification	200
A.2.18	LiteralString	202
A.2.19	LiteralUnlimitedNatural	204
A.2.20	MultiplicityElement	206
A.2.21	NamedElement	211
A.2.22	Namespace	214
A.2.23	OpaqueExpression	217
A.2.24	Operation	219
A.2.25	Package	225
A.2.26	PackageableElement	229
A.2.27	PackageImport	231
A.2.28	Parameter	235
A.2.29	ParameterDirectionKind	238
A.2.30	Property	240
A.2.31	RedefinableElement	248
A.2.32	Relationship	251
A.2.33	Slot	253
A.2.34	StructuralFeature	255
A.2.35	Type	257
A.2.36	TypedElement	259
A.2.37	ValueSpecification	261
A.2.38	VisibilityKind	263
A.3	Die PML Prozessklassen	266
A.3.1	Analogien zu MOF/UML	266
A.3.2	Activity	269
A.3.3	Event	275
A.3.4	EventParameter	279
A.3.5	Information	282
A.3.6	ProcessClass	284
A.3.7	ProcessProperty	288
A.3.8	Resource	291
A.4	Diagramme	296
A.4.1	Klassen- und Paketdiagramme	296
A.4.2	Das Aktivitätsdiagramm	297