



Universiteit
Leiden
The Netherlands

Domain specific modeling and analysis

Jacob, J.F.

Citation

Jacob, J. F. (2008, November 13). *Domain specific modeling and analysis*. Retrieved from <https://hdl.handle.net/1887/13257>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/13257>

Note: To cite this publication please use the final published version (if applicable).

Domain Specific Modeling and Analysis

PROEFSCHRIFT

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. P.F. van der Heijden,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 13 november 2008
klokke 15.00 uur

door

Joost Ferdinand Jacob

geboren te Haarlem
in 1963

Promotiecommissie

Promotor: Prof. Dr. F.S. de Boer

Co-promotor: Dr. M.M. Bonsangue

Referent: Prof. Dr. H.A. Proper
University of Nijmegen

Overige leden: Prof. Dr. J.N. Kok

Prof. Dr. F. Arbab

Prof. Dr. W.-P. de Roever
Christian-Albrechts-University of Kiel, Germany

The work reported in this thesis has been carried out at the Center for Mathematics and Computer Science (CWI) in Amsterdam.

Contents

1	Introduction	1
1.1	Problem statement	3
1.1.1	The OMEGA project	3
1.1.2	The Archimate project	5
1.2	Objectives	8
1.3	Approach	9
1.4	Working with XML and other structured data	12
1.5	Structure of the thesis	13
1.6	Conclusion	16
I	RML, a tool for model analysis	21
2	RML	23
2.1	Introduction	23
2.2	XML and XML transformations	25
2.2.1	XSLT	27
2.3	RML	28
2.3.1	XML Wildcard Elements	29
2.3.2	The RML syntax	29
2.3.3	The RML tools and libraries	31
2.4	RML examples	32
2.4.1	Executable UML models	32
2.4.2	Other examples	37
2.5	Related work and conclusion	37
3	The RML Tutorial	41
3.1	The XML vocabulary for the examples	42

3.2	How RML works	43
3.2.1	Rules	43
3.2.2	Literal matching	44
3.2.3	Wildcard matching	44
3.2.4	Search and replace	45
3.2.5	The dorules tool	45
3.2.6	The dorecipe tool	46
3.2.7	XML parsing details	47
3.3	Future versions of RML	47
3.4	Installing and running	49
3.5	Examples	51
3.5.1	Deleting an element	51
3.5.2	Changing an element	53
3.5.3	RML variables for elements	55
3.5.4	RML variables for text content	58
3.5.5	Adding constraints with rml-if	59
3.5.6	Match choice with rml-type="or"	66
3.5.7	How to remove duplicate siblings	67
3.5.8	Iterating sets of rules	69
3.5.9	Turning a list into a hierarchy	73
3.5.10	Pre-binding string variables on the commandline	74
3.5.11	Using recipes	75
 II Component Models and Analysis		77
4	The OMEGA Component Model	79
4.1	Introduction	79
4.2	The Component Model	80
4.2.1	Blackbox Components	83
4.2.2	Basic Components	86
4.2.3	Extensions to the OMEGA UML subset discussed so far (Fig. 4.1)	86
4.2.4	Component Systems	89
4.3	Runtime Behaviour	93
4.4	Architectural Models	94
4.4.1	Overview	98
4.5	Trace Semantics	99

4.6	Modeling with Components	101
4.6.1	Examples of software developed with the component model	102
4.7	Conclusion and related work	103
5	Component Coordination in UML	107
5.1	Introduction	107
5.2	A component model	109
5.3	Ω -UML	116
5.3.1	Components in Ω -UML	117
5.4	Intra-component coordination	119
5.5	Inter-components coordination	121
5.6	Conclusion	123
6	UnCL	125
6.1	Introduction	125
6.2	Semantics of UnCL	127
6.3	The UnCL Execution Platform	131
6.4	UnCL and Mobile Channels	136
6.4.1	MoCha's Mobile Channels	136
6.4.2	Channel Types	138
6.4.3	Implementation	138
6.4.4	UnCL and MoCha	138
6.5	Conclusions and Related Work	140
7	ATL	143
7.1	Introduction	143
7.2	ASCII Transformation Language (ATL)	145
7.2.1	Preliminary: λ -calculus	146
7.2.2	ATL	148
7.2.3	Implementation	151
7.2.4	Definition of the γ -reduction	151
7.3	A webapplication	152
7.3.1	Equivalence classes and conflict relations	155
7.3.2	Adding user-defined rules	156
7.4	The Sieve example	157
7.5	Related work and the future	159

III	Modeling and Analysing Architectures	161
8	Analysis with XML	163
8.1	Introduction	163
8.2	ArchiMate: a running example	164
8.3	The tools: XML, AML and RML	167
8.4	Static analysis	171
8.4.1	A formal basis for static analysis	172
8.4.2	XML for static analysis	177
8.5	Dynamic analysis	180
8.5.1	A formal basis for dynamic analysis	180
8.5.2	XML+RML for dynamic analysis	182
8.6	Summary and outlook	187
9	A Logical Viewpoint	189
9.1	Introduction	189
9.2	Archimate: a running example	192
9.2.1	Systems and architectures	193
9.3	Semantic models	199
9.4	Design support	203
9.5	Tool support	205
9.5.1	The Rule Markup Language	205
9.5.2	RML as a tool for architectural description	207
9.6	Summary and outlook	208
10	Transformations	211
10.1	Introduction	211
10.1.1	Enterprise Architectures	211
10.1.2	ArchiMate	212
10.1.3	XML	212
10.1.4	Research methodology	213
10.1.5	Document layout	214
10.2	The Rule Markup Language	214
10.2.1	Comparison with other techniques	215
10.3	Running Example	216
10.3.1	An XML description of the example	218
10.4	Selection and Visualisation	219
10.4.1	Selection	219

10.4.2 Visualization	220
10.5 Analysis	222
10.6 Summary	223
10.6.1 Question 1	224
10.6.2 Question 2	224
10.6.3 Question 3	225
10.6.4 Conclusions	225
Summary	237
Samenvatting	238
Curriculum Vitae	239

