Structures and Processes for Managing Model-Metamodel Co-evolution

Louis Mathew Rose

This thesis is submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy.

University of York,
York,
YO10 5DD

Department of Computer Science

October 2010

Abstract

Software changes over time. During the lifetime of a software system, unintended behaviour must be corrected and new requirements satisfied. Because software changes are costly, tools for automatically managing change are commonplace. Contemporary development environments can automatically perform change management tasks such as impact analysis, refactoring and background compilation.

Increasingly, models and modelling languages are first-class citizens in software development. Model-Driven Engineering (MDE), a state-of-the-art approach to software engineering, prescribes the use of models throughout the software engineering process and uses automated transformations to generate code from models.

Contemporary MDE environments provide little support for managing a type of evolution termed *model-metamodel co-evolution*, in which changes to a modelling language are propagated to models. This thesis demonstrates that model-metamodel co-evolution occurs often in MDE projects, and that dedicated structures and processes for its management increase the productivity and understandability of the development process. Structures and processes for managing model-metamodel co-evolution are proposed, developed, and then evaluated by comparison to existing structures and processes with quantitive and qualitative techniques.

For my Nanna Spence

Contents

Contents **vii**

List of Figures **ix**

List of Tables **xii**

Listings **xiii**

1 Introduction **1**
- 1.1 Model-Driven Engineering 2
- 1.2 Software Evolution 3
- 1.3 Motivation: Software Evolution in MDE 4
- 1.4 Research Hypothesis 4
- 1.5 Research Method 6
- 1.6 Research Results 8
- 1.7 Thesis Structure 8

2 Background **11**
- 2.1 MDE Terminology and Principles 11
- 2.2 MDE Guidelines and Methods 22
- 2.3 MDE Tools 27
- 2.4 Research Relating to MDE 33
- 2.5 Benefits of and Current Challenges for MDE 36
- 2.6 Chapter Summary 39

3 Literature Review **41**
- 3.1 Software Evolution Theory 41
- 3.2 Software Evolution in Practice 46
- 3.3 Research Challenges 65
- 3.4 Chapter Summary 66

4 Analysis **69**
- 4.1 Locating Data 70
- 4.2 Analysing Existing Techniques 77

4.3 Requirements Identification 88
4.4 Chapter Summary 91

5 Implementation 93
5.1 Metamodel-Independent Syntax 93
5.2 Textual Modelling Notation 99
5.3 Analysis of Languages used for Migration 111
5.4 Epsilon Flock: A Model Migration Language 119
5.5 Chapter Summary 130

6 Evaluation 133
6.1 Evaluating User-Driven Co-Evolution 134
6.2 Evaluating Conservative Copy 149
6.3 Evaluating Co-evolution Tools 169
6.4 Transformation Tools Contest 185
6.5 Limitations 200
6.6 Chapter Summary 201

7 Conclusions 201
7.1 Research Contributions 202
7.2 Future Work 205
7.3 Coda 207

A A Graphical Editor for Process-Oriented Programs 209
A.1 Iteration 1: Processes and Channels 210
A.2 Iteration 2: Interoperability with GMF 211
A.3 Iteration 3: Shared Channels 213
A.4 Iteration 4: Connection Points 215
A.5 Iteration 5: Connection Point Types 219
A.6 Iteration 6: Nested Processes and Channels 222
A.7 Summary 224

B Co-evolution Examples 225
B.1 Newsgroups Examples 225
B.2 UML Example 230
B.3 GMF Examples 236

C TTC Results 281

Bibliography 287

List of Figures

1.1 Overview of the research method. 7

2.1 Jackson's definition of a model 12
2.2 A fragment of the UML metamodel defined in MOF 15
2.3 Exemplar State Machine metamodel. 17
2.4 Exemplar Object-Oriented metamodel. 18
2.5 Interactions between a PIM and several PSMs. 23
2.6 The tiers of standards used as part of MDA. 23
2.7 An EMF model editor for state machines. 29
2.8 EMF's tree-based metamodel editor. 29
2.9 EMF's graphical metamodel editor. 30
2.10 The Emfatic textual metamodel editor for EMF. 31
2.11 GMF state machine model editor. 31
2.12 The architecture of Epsilon 32

3.1 Categories of traceability link 46
3.2 Attribute to association end refactoring in EMF Refactor 56
3.3 Approaches to incremental transformation 57
3.4 Exemplar impact analysis pattern 59
3.5 An exemplar co-evolution process 62
3.6 Visualising a transformation chain 64

4.1 Analysis chapter overview. 69
4.2 Refactoring a reference to a value 75
4.3 Co-evolution activities 78
4.4 Metamodel evolution in the Epsilon FPTC tool 85
4.5 Spectrum of developer-driven co-evolution approaches 88

5.1 Implementation chapter overview. 93
5.2 A generic metamodel. 95
5.3 Minimal MOF metamodel. 95
5.4 Exemplar instantiation of generic metamodel. 97
5.5 Exemplar families metamodel 100
5.6 The architecture of Epsilon HUTN. 104

5.7 Conformance problem reporting in Epsilon HUTN. 109
5.8 Exemplar metamodel evolution (Petri nets) 112
5.9 Mappings between the original and evolved Petri nets metamodels 115
5.10 The metamodel-independent representation used by COPE 117
5.11 The abstract syntax of Flock. 120
5.12 Exemplar Process-Oriented metamodel evolution 124
5.13 Exemplar Process-Oriented model prior to migration 125
5.14 Exemplar UML metamodel evolution 129

6.1 Final version of the prototypical graphical model editor. 137
6.2 The graphical editor at the start of the iteration. 138
6.3 The graphical editor at the end of the iteration. 139
6.4 Process-oriented metamodel evolution. 140
6.5 User-driven co-evolution with EMF 141
6.6 XMI prior to migration 142
6.7 XMI after migration 143
6.8 User-driven co-evolution with dedicated structures 145
6.9 HUTN source prior to migration 146
6.10 HUTN source part way through migration 147
6.11 Exemplar metamodel evolution (Petri nets) 159
6.12 Simplified fragment of the GMF Graph metamodel. 163
6.13 Change Reference to Containment metamodel evolution 167
6.14 Exemplar metamodel evolution (Petri nets) 171
6.15 GMF graph metamodel evolution 173
6.16 Migration tool performance comparison. 182
6.17 Exemplar activity model. 187
6.18 UML 1.4 Activity Graphs 188
6.19 UML 2.2 Activity Diagrams 189
6.20 Migrating Actions for the Core Task 190
6.21 Migrating Actions for Extension 1 191

A.1 The process-oriented metamodel after one iteration. 210
A.2 The process-oriented metamodel after two iterations. 211
A.3 Exemplar diagram after the second iteration. 212
A.4 The process-oriented metamodel after three iterations. 213
A.5 Exemplar migration between the second and third versions of the process-oriented metamodel 214
A.6 The process-oriented metamodel after the fourth iteration. 215
A.7 Exemplar diagram after the fourth iteration. 217
A.8 Exemplar migration between the third and fourth versions of the process-oriented metamodel 218
A.9 The process-oriented metamodel after five iterations. 219
A.10 Exemplar migration between the fourth and fifth versions of the process-oriented metamodel 221

A.11 The process-oriented metamodel after six iterations. 222
A.12 Exemplar diagram after the final iteration. 224

B.1 Newsgroups metamodel during the Extract Person iteration 226
B.2 Newsgroups metamodel during the Resolve Replies iteration 228
B.3 Activities in UML 1.4 and UML 2.2 231
B.4 The Graph metamodel in GMF 1.0 and GMF 2.0 238

List of Tables

4.1 Candidates for study of evolution in existing MDE projects 71

5.1 Properties of model migration approaches 129

6.1 Model operation frequency (analysis examples). 160
6.2 Model operation frequency (evaluation examples). 160
6.3 Summary of comparison criteria. 174
6.4 Summary of tool selection advice 183
6.5 TTC scores for Epsilon Flock (unweighted). 197

C.1 Correctness scores 282
C.2 Conciseness scores 282
C.3 Clarity scores 283
C.4 Appropriateness scores 283
C.5 Tool maturity scores 284
C.6 Reproducibility scores 284
C.7 Extensions scores 285
C.8 Total (equally weighted) scores 285
C.9 Total (weighted) scores 286

Listings

2.1 Exemplar M2M transformation in ETL 17
2.2 Exemplar M2T transformation in EGL 19
2.3 Exemplar T2M transformation in EMFtext 20
2.4 Exemplar model validation in EVL 21
4.1 Migration strategy for the refactoring in pseudo code. 75
5.1 Exemplar person model in XMI 94
5.2 Specifying attributes with HUTN. 101
5.3 Specifying a containment reference with HUTN. 101
5.4 Specifying a simple reference with HUTN. 101
5.5 Using keywords and adjectives in HUTN. 102
5.6 Referencing objects in other packages with HUTN. 102
5.7 Using a reference block in HUTN. 103
5.8 Using an infix reference in HUTN. 103
5.9 Transforming Nodes to PackageObjects with ETL. 105
5.10 A constraint (in EVL) to check that all identifiers are unique. . 106
5.11 Higher-order transformation with EGL 107
5.12 The M2M transformation generated for the Families metamodel 108
5.13 HUTN for people with mothers and fathers. 109
5.14 HUTN for people with parents. 110
5.15 Failure behaviour specified in HUTN. 110
5.16 Fragment of the Petri nets model migration in ATL 114
5.18 Petri nets model migration in COPE 117
5.19 Concrete syntax of migrate and delete rules. 121
5.20 Redefining equivalences for the Component model migration. . 126
5.21 Petri nets model migration in Flock 127
5.22 UML model migration in Flock 127
6.1 Assignment operators in ATL 154
6.2 The Petri nets model migration in ATL 157
6.3 The Petri nets model migration in Groovy-for-COPE 158
6.4 Petri nets model migration in Flock 158
6.5 An extract of the GMF Graph model migration in ATL 162
6.6 Simplified GMF Graph model migration in ATL 165
6.7 Simplified GMF Graph model migration in COPE 165
6.8 Simplified GMF Graph model migration in Flock 166

6.9 Migration for Change Reference to Containment in ATL 168
6.10 Migration for Change Reference to Containment in Flock . . . 169
6.11 Migrating Actions 193
6.12 Migrating FinalStates and Transitions 193
6.13 Migrating Pseudostates 193
6.14 Migrating ActivityGraphs 193
6.15 Migrating Guards 194
6.16 Migrating Partitions 194
6.17 Migrating ObjectFlows 195
6.18 Migrating ObjectFlowStates to a single ObjectFlow 195
6.19 Migrating Partitions without ObjectFlowStates 195
A.1 The annotated process-oriented metamodel after one iteration . 210
A.2 The annotated process-oriented metamodel after two iterations 211
A.3 The annotated process-oriented metamodel after four iterations 215
A.4 The annotated process-oriented metamodel after five iterations 219
A.5 The annotated process-oriented metamodel after six iterations . 222
B.1 The Newsgroup Extract Person model migration in ATL 226
B.2 The Newsgroup Extract Person model migration in Groovy-for-COPE 227
B.3 The Newsgroup Extract Person model migration in Flock . . . 227
B.4 The Newsgroup Resolve Replies model migration in ATL . . . 228
B.5 The Newsgroup Resolve Replies model migration in Groovy-for-COPE 229
B.6 The Newsgroup Resolve Replies model migration in Flock . . . 229
B.7 UML activity diagram model migration in ATL 230
B.8 UML activity diagram model migration in Groovy-for-COPE . 234
B.9 UML activity diagram model migration in Flock 235
B.10 GMF Graph model migration in ATL 237
B.11 GMF Graph model migration in Groovy-for-COPE 249
B.12 GMF Graph model migration in Flock 250
B.13 GMF Generator model migration in ATL 251
B.14 GMF Generator model migration in Groovy-for-COPE 276
B.15 GMF Generator model migration in Flock 278

Acknowledgements

To be completed.

Author Declaration

Except where stated, all of the work contained in this thesis represents the original contribution of the author. Section 6.3 reports collaborative experiments with model migration tools, and that section makes clear the roles of the author and other participants.

Parts of the work described in this thesis have been previously published by the author in:

- **The Epsilon Generation Language**, Louis M. Rose and Richard F. Paige and Dimitrios S. Kolovos and Fiona A.C. Polack in *Proc. European Conference on Model Driven Architecture – Foundations and Applications (ECMDA-FA)*, volume 5095 of LNCS, pages 1-16. Springer, 2008.
- **Constructing Models with the Human-Usable Textual Notation**, Louis M. Rose and Richard F. Paige and Dimitrios S. Kolovos and Fiona A.C. Polack in *Proc. International Conference on Model Driven Engineering Languages and Systems (MoDELS)*, volume 5301 of LNCS, pages 249-263. Springer, 2008.
- **An Analysis of Approaches to Model Migration**, Louis M. Rose and Richard F. Paige and Dimitrios S. Kolovos and Fiona A.C. Polack in *Proc. Joint Model-Driven Software Evolution and Model Co-evolution and Consistency Management (MoDSE-MCCM) Workshop*, co-located with MoDELS 2009.
- **Enhanced Automation for Managing Model and Metamodel Inconsistency**, Louis M. Rose and Dimitrios S. Kolovos and Richard F. Paige and Fiona A.C. Polack in *Proc. International Conference on Automated Software Engineering (ASE)*, pages 545-549, ACM Press, 2009.
- **Concordance: An Efficient Framework for Managing Model Integrity**, Louis M. Rose, Dimitrios S. Kolovos, Nicholas Drivalos, James. R. Williams, Richard F. Paige, Fiona A.C. Polack, and Kiran J. Fernandes in *Proc. European Conference on Modelling Foundations and Applications*

(ECMFA), volume 6138 of LNCS, pages 62-73. Springer, 2010.

- **Model Migration with Epsilon Flock**, Louis M. Rose, Dimitrios S. Kolovos, Richard F. Paige, and Fiona A.C. Polack in *Proc. International Conference on the Theory and Practice of Model Transformations (ICMT)*, volume 6142 of LNCS, pages 184-198. Springer, 2010.
- **Model Migration Case**, Louis M. Rose, Dimitrios S. Kolovos, Richard F. Paige, and Fiona A.C. Polack in *Proc. Transformation Tools Contest (TTC)*, co-located with TOOLS 2010.
- **Migrating Activity Diagrams with Epsilon Flock**, Louis M. Rose, Dimitrios S. Kolovos, Richard F. Paige, and Fiona A.C. Polack in *Proc. Transformation Tools Contest (TTC)*, co-located with TOOLS 2010.

In addition, the author has contributed to [Kolovos *et al.* 2007a], [Kolovos *et al.* 2007b] and [Paige *et al.* 2009].

Bibliography

[Ackoff 1962] R.L. Ackoff. *Scientific Method: Optimizing Applied Research Decisions*. John Wiley and Sons, New York, 1962.

[Aizenbud-Reshef *et al.* 2005] N. Aizenbud-Reshef, R.F. Paige, J. Rubin, Y. Shaham-Gafni, and D.S. Kolovos. Operational semantics for traceability. In *Proc. Workshop on Traceability, co-located with the European Conference on Model-Driven Architecture (ECMDA)*, pages 8–14, 2005.

[Alexander *et al.* 1977] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*. Oxford University Press, New York, 1977.

[Álvarez *et al.* 2001] J. Álvarez, A. Evans, and P. Sammut. MML and the metamodel architecture. In *Proc. Workshop on Transformation in UML, co-located with the European Joint Conferences on Theory and Practice of Software (ETAPS)*, 2001.

[Amyot *et al.* 2006] D. Amyot, H. Farah, and J.-F. Roy. Evaluation of development tools for domain-specific modeling languages. In R. Gotzhein and R. Reed, editors, *System Analysis and Modeling: Language Profiles*, volume 4320 of *Lecture Notes in Computer Science*, pages 183–197. Springer, Heidelberg, 2006.

[Apostel 1960] L. Apostel. Towards the formal study of models in the non-formal sciences. *Synthese*, 12(2):125–161, 1960.

[Arendt *et al.* 2009] T. Arendt, F. Mantz, L. Schneider, and G. Taentzer. Model refactoring in Eclipse by LTK, EWL, and EMF Refactor: A case study. In *Proc. Joint Model-Driven Software Evolution and Model Co-evolution and Consistency Management (MoDSE-MCCM) Workshop, co-located with the International Conference on Model-Driven Engineering Languages and Systems (MoDELS)*, 2009.

[Backus 1978] J. Backus. The history of FORTRAN I, II and III. *History of Programming Languages*, 1:165–180, 1978.

[Balazinska *et al.* 2000] M. Balazinska, E. Merlo, M. Dagenais, B. Lagüe, and K. Kontogiannis. Advanced clone-analysis to support object-oriented system refactoring. In *Proc. Working Conference on Reverse Engineering (WCRE)*, pages 98–107. IEEE Computer Society, 2000.

[Banerjee *et al.* 1987] J. Banerjee, W. Kim, H. Kim, and H.F. Korth. Semantics and implementation of schema evolution in object-oriented databases. In U. Dayal and I.L. Traiger, editors, *Proc. Special Interest Group on Management of Data (SIGMOD) Conference*, pages 311–322. ACM Press, 1987.

[Beck & Cunningham 1989] K. Beck and W. Cunningham. Constructing abstractions for object-oriented applications. *Journal of Object Oriented Programming*, 2, 1989.

[Bézivin & Gerbé 2001] J. Bézivin and O. Gerbé. Towards a precise definition of the OMG/MDA framework. In *Proc. International Conference on Automated Software Engineering (ASE)*, pages 273–280. IEEE Computer Society, 2001.

[Bézivin 2005] J. Bézivin. On the unification power of models. *Software and System Modeling*, 4(2):171–188, 2005.

[Biermann *et al.* 2006] E. Biermann, K. Ehrig, C. Köhler, G. Kuhns, G. Taentzer, and E. Weiss. EMF model refactoring based on graph transformation concepts. *Electronic Communications of the European Association for the Study of Science and Technology [online]*, 3, 2006. Available at `http://eceasst.cs.tu-berlin.de/index.php/eceasst/article/view/34` [Accessed 2 November 2010].

[Bloch 2005] J. Bloch. How to design a good API and why it matters [online]. Keynote address to the LCSD Workshop at OOPSLA, October 2005, San Diego, United States of America. Available at: `http://lcsd05.cs.tamu.edu/slides/keynote.pdf` [Accessed 2 November 2010], 2005.

[Bohner 2002] S.A. Bohner. Software change impacts - an evolving perspective. In *Proc. International Conference on Software Maintenance (ICSM)*, pages 263–272. IEEE Computer Society, 2002.

[Bosch 1998] J. Bosch. Design patterns as language constructs. *Journal of Object Oriented Programming*, 11(2):18–32, 1998.

[Briand *et al.* 2003] L.C. Briand, Y. Labiche, and L. O'Sullivan. Impact analysis and change management of uml models. In *Proc. International Conference on Software Maintenance (ICSM)*, pages 256–265. IEEE Computer Society, 2003.

[Brooks Jr. 1987] F.P. Brooks Jr. No silver bullet - essence and accidents of software engineering. *IEEE Computer*, 20(4):10–19, 1987.

[Brown *et al.* 1998] W.J. Brown, R.C. Malveau, H.W. McCormick III, and T.J. Mowbray. *Anti Patterns*. Wiley, New York, 1998.

[Cervelle *et al.* 2006] J. Cervelle, R. Forax, and G. Roussel. Tatoo: an innovative parser generator. In Ralf Gitzel, Markus Aleksy, and Martin Schader, editors, *Proc. International Symposium on Principles and Practice of Programming in Java (PPPJ)*, volume 178 of *ACM International Conference Proceeding Series*, pages 13–20. ACM, 2006.

[Chen & Chou 1999] J.Y.J. Chen and S.C. Chou. Consistency management in a process environment. *Systems and Software*, 47(2-3):105–110, 1999.

[Cicchetti *et al.* 2008] A. Cicchetti, D. Di Ruscio, R. Eramo, and A. Pierantonio. Automating co-evolution in model-driven engineering. In *Proc. International IEEE Enterprise Distributed Object Computing Conference (EDOC)*, pages 222–231. IEEE Computer Society, 2008.

[Cicchetti 2008] A. Cicchetti. *Difference Representation and Conflict Management in Model-Driven Engineering*. PhD thesis, Universita' degli Studi dell'Aquila, L'Aquila, Italy, 2008.

[Clark *et al.* 2008] T. Clark, P. Sammut, and J. Willians. *Superlanguages: Developing Languages and Applications with XMF [online]*. Ceteva, Sheffield, 2008. Available at: `http://itcentre.tvu.ac.uk/~clark/Papers/Superlanguages.pdf` [Accessed 02 November 2010].

[Cleland-Huang *et al.* 2003] J. Cleland-Huang, C.K. Chang, and M. Christensen. Event-based traceability for managing evolutionary change. *IEEE Transactions on Software Engineering*, 29(9):796–810, 2003.

[Costa & Silva 2007] M. Costa and A.R. da Silva. RT-MDD framework – a practical approach. In *Proc. Workshop on Traceability, co-located with the European Conference on Model-Driven Architecture (ECMDA)*, pages 17–26, 2007.

[Czarnecki & Helsen 2006] K. Czarnecki and S. Helsen. Feature-based survey of model transformation approaches. *IBM Systems Journal*, 45(3):621–646, 2006.

[Deursen *et al.* 2000] A. van Deursen, P. Klint, and J. Visser. Domain-Specific Languages: An annotated bibliography. *Special Interest Group on Programming Languages (SIGPLAN) Notices*, 35(6):26–36, 2000.

[Deursen *et al.* 2007] A. van Deursen, E. Visser, and J. Warmer. Model-driven software evolution: A research agenda. In *Proc. Workshop on Model-Driven Software Evolution, co-located with the European Conference on Software Maintenance and Reengineering (CSMR)*, pages 41–49, 2007.

[Dig & Johnson 2006a] D. Dig and R. Johnson. Automated upgrading of component-based applications. In P.L. Tarr and W.R. Cook, editors, *Companion to the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, pages 675–676, 2006.

[Dig & Johnson 2006b] D. Dig and R. Johnson. How do APIs evolve? A story of refactoring. *Journal of Software Maintenance and Evolution*, 18(2):83–107, 2006.

[Dig *et al.* 2006] D. Dig, C. Comertoglu, D. Marinov, and R. Johnson. Automated detection of refactorings in evolving components. In D. Thomas, editor, *Proc. European Conference on Object-Oriented Programming (ECOOP)*, volume 4067 of *Lecture Notes in Computer Science*, pages 404–428. Springer, 2006.

[Dig *et al.* 2007] D. Dig, K. Manzoor, R. Johnson, and T.N. Nguyen. Refactoring-aware configuration management for object-oriented programs. In *Proc. International Conference on Software Engineering (ICSE)*, pages 427–436. IEEE Computer Society, 2007.

[Dig 2007] D. Dig. *Automated Upgrading of Component-Based Applications.* PhD thesis, University of Illinois at Urbana-Champaign, USA, 2007.

[Drivalos *et al.* 2008] N. Drivalos, R.F. Paige, K.J. Fernandes, and D.S. Kolovos. Towards rigorously defined model-to-model traceability. In *Proc. Workshop on Traceability, co-located with the European Conference on Model Driven Architecture (ECMDA)*, 2008.

[Ducasse *et al.* 1999] S. Ducasse, M. Rieger, and S. Demeyer. A language independent approach for detecting duplicated code. In *Proc. International Conference on Software Maintenance (ICSM)*, pages 109–118. IEEE Computer Society, 1999.

[Edelweiss & Moreira 2005] N. Edelweiss and Á.F. Moreira. Temporal and versioning model for schema evolution in object-oriented databases. *Data & Knowledge Engineering*, 53(2):99–128, 2005.

[Elmasri & Navathe 2006] R. Elmasri and S.B. Navathe. *Fundamentals of Database Systems.* Addison-Wesley Longman, Boston, Massachusetts, 5th edition, 2006.

[Erlikh 2000] L. Erlikh. Leveraging legacy system dollars for e-business. *IT Professional*, 2(3):17–23, 2000.

[Evans 2004] E. Evans. *Domain-Driven Design: Tacking Complexity In the Heart of Software*. Addison-Wesley, Boston, Massachusetts, 2004.

[Feathers 2004] M.C. Feathers. *Working Effectively with Legacy Code*. Prentice Hall, Upper Saddle River, New Jersey, 2004.

[Ferrandina *et al.* 1995] F. Ferrandina, T. Meyer, R. Zicari, G. Ferran, and J. Madec. Schema and database evolution in the O2 object database system. In U. Dayal, P.M.D. Gray, and S. Nishio, editors, *Proc. International Conference on Very Large Data Bases (VLDB)*, pages 170–181. Morgan Kaufmann, 1995.

[Fowler 1999] M. Fowler. *Refactoring: improving the design of existing code*. Addison-Wesley, Upper Saddle River, New Jersey, 1999.

[Fowler 2002] M. Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, Massachusetts, 2002.

[Fowler 2010] M. Fowler. *Domain Specific Languages*. Addison-Wesley Professional, Boston, Massachusetts, 2010.

[Frankel 2002] D. Frankel. *Model Driven Architecture: Applying MDA to Enterprise Computing*. John Wiley & Sons Inc., New York, 2002.

[Fritzsche *et al.* 2008] M. Fritzsche, J. Johannes, S. Zschaler, A. Zherebtsov, and A. Terekhov. Application of tracing techniques in model-driven performance engineering. In *Proc.Traceability Workshop, co-located with the European Conference on Model-Driven Architecture (ECMDA)*, pages 111–120, 2008.

[Fuhrer *et al.* 2007] R.M. Fuhrer, A. Kiezun, and M. Keller. Refactoring in the Eclipse JDT: Past, present, and future. In *Proc. Workshop on Refactoring Tools (WRT), co-located with European Conference on Object-Oriented Programming (ECOOP)*, 2007.

[Gamma *et al.* 1995] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, Reading, Massachusetts, 1995.

[Garcés *et al.* 2009] K. Garcés, F. Jouault, P. Cointe, and J. Bézivin. Managing model adaptation by precise detection of metamodel changes. In R.F. Paige, A. Hartman, and A. Rensink, editors, *Proc. European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA)*, volume 5562 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2009.

[Geiß & Kroll 2007] R. Geiß and M. Kroll. Grgen.net: A fast, expressive, and general purpose graph rewrite tool. In A. Schürr, M. Nagl, and A. Zündorf, editors, *Proc. International Symposium on Applications of Graph Transformations with Industrial Relevance (AGTIVE), Revised Selected and Invited Papers*, volume 5088 of *Lecture Notes in Computer Science*, pages 568–569. Springer, 2007.

[Gosling *et al.* 2005] J. Gosling, B. Joy, G. Steele, and G. Bracha. *The Java^{TM} Language Specification*. Addison-Wesley, Boston, Massachusetts, 2005.

[Graham 1993] P. Graham. *On Lisp: Advanced Techniques for Common Lisp*. Prentice-Hall, Upper Saddle River, New Jersey, 1993.

[Greenfield *et al.* 2004] J. Greenfield, K. Short, S. Cook, and S. Kent. *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. John Wiley & Sons Inc., New York, 2004.

[Gronback 2009] R.C. Gronback. *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. Addison-Wesley Professional, Boston, Massachusetts, 2009.

[Gruschko *et al.* 2007] B. Gruschko, D.S. Kolovos, and R.F. Paige. Towards synchronizing models with evolving metamodels. In *Proc. Workshop on Model-Driven Software Evolution, co-located with the European Conference on Software Maintenance and Reengineering (CSMR)*, 2007.

[Guerrini *et al.* 2005] G. Guerrini, M. Mesiti, and D. Rossi. Impact of XML schema evolution on valid documents. In A. Bonifati and D. Lee, editors, *Proc. International Workshop on Web Information and Data Management (WIDM),*, pages 39–44. ACM, 2005.

[Halstead 1977] M.H. Halstead. *Elements of Software Science*. Elsevier Science Inc., New York, 1977.

[Hearnden *et al.* 2006] D. Hearnden, M. Lawley, and K. Raymond. Incremental model transformation for the evolution of model-driven systems. In O. Nierstrasz, J. Whittle, D. Harel, and G. Reggio, editors, *Proc. International Conference on Model Driven Engineering Languages and Systems (MoDELS)*, volume 4199 of *Lecture Notes in Computer Science*, pages 321–335. Springer, 2006.

[Heidenreich *et al.* 2009] F. Heidenreich, J. Johannes, S. Karol, M. Seifert, and C. Wende. Derivation and refinement of textual syntax for models. In R.F. Paige, A. Hartman, and A. Rensink, editors, *Proc. European Conference on Model-Driven Architecture - Foundations and Applications (ECMDA-FA)*, volume 5562 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2009.

[Herrmannsdoerfer *et al.* 2008] M. Herrmannsdoerfer, S. Benz, and E. Juergens. Automatability of coupled evolution of metamodels and models in practice. In K. Czarnecki, I. Ober, J. Bruel, A. Uhl, and M. Völter, editors, *Proc. International Conference on Model Driven Engineering Languages and Systems (MoDELS)*, volume 5301 of *LNCS*, pages 645–659. Springer, 2008.

[Herrmannsdoerfer *et al.* 2009a] M. Herrmannsdoerfer, S. Benz, and E. Juergens. COPE - automating coupled evolution of metamodels and models. In S. Drossopoulou, editor, *Proc. European Conference on Object-Oriented Programming (ECOOP)*, volume 5653 of *Lecture Notes in Computer Science*, pages 52–76. Springer, 2009.

[Herrmannsdoerfer *et al.* 2009b] M. Herrmannsdoerfer, D. Ratiu, and G. Wachsmuth. Language evolution in practice. In M. van den Brand, D. Gasevic, and J. Gray, editors, *Proc. International Conference on Software Language Engineering (SLE), Revised Selected Papers*, volume 5696 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2009.

[Hussey & Paternostro 2006] K. Hussey and M. Paternostro. Advanced features of EMF. Tutorial at EclipseCon 2006, California, USA. [Accessed 07 September 2009] Available at: `http://www.eclipsecon.org/2006/Sub.do?id=171`, 2006.

[ISO/IEC 1996] Information Technology ISO/IEC. Syntactic metalanguage – Extended BNF. ISO 14977:1996 International Standard, 1996.

[ISO/IEC 2002] Information Technology ISO/IEC. Z Formal Specification Notation – Syntax, Type System and Semantics. ISO 13568:2002 International Standard, 2002.

[Jackson 1995] M. Jackson. *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices.* ACM Press, New York, 1995.

[JetBrains 2008] JetBrains. MPS – Meta Programming System [online]. [Accessed 30 June 2008] Available at: `http://www.jetbrains.com/mps/index.html`, 2008.

[Jouault & Kurtev 2005] F. Jouault and I. Kurtev. Transforming models with ATL. In *Proc. Satellite Events at the International Conference on Model Driven Engineering Languages and Systems*, volume 3844 of *LNCS*, pages 128–138. Springer, 2005.

[Jouault 2005] F. Jouault. Loosely coupled traceability for ATL. In *Proc. ECMDA-FA Workshop on Traceability*, 2005.

[Jurack & Mantz 2010] S. Jurack and F. Mantz. Towards metamodel evolution of EMF models with Henshin. In *Proc. ME Workshop*, 2010.

[Kalnins *et al.* 2005] A. Kalnins, J. Barzdins, and E. Celms. Model transformation language MOLA. In *Proc. Model Driven Architecture, European MDA Workshops: Foundations and Applications MDAFA*, volume 3599 of *Lecture Notes in Computer Science*, pages 62–76. Springer, 2005.

[Kataoka *et al.* 2001] Y. Kataoka, M.D. Ernst, W.G. Griswold, and D. Notkin. Automated support for program refactoring using invariants. In *Proc. International Conference on Software Maintenance*, pages 736–743. IEEE Computer Society, 2001.

[Kelly & Tolvanen 2008] S. Kelly and J.P. Tolvanen. *Domain-Specific Modelling.* Wiley and IEEE Computer Society, 2008.

[Kerievsky 2004] J. Kerievsky. *Refactoring to Patterns.* Pearson Higher Education, 2004.

[Kleppe *et al.* 2003] A.G. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise.* Addison-Wesley Longman Publishing Co. Inc., Boston, Massachusetts, 2003.

[Klint *et al.* 2003] P. Klint, R. Lämmel, and C. Verhoef. Towards an engineering discipline for grammarware. *ACM Transactions on Software Engineering Methodology*, 14:331–380, 2003.

[Kolovos *et al.* 2006a] D.S. Kolovos, R.F. Paige, and F.A. Polack. The Epsilon Object Language (EOL). In *Proc. ECMDA-FA*, volume 4066 of *LNCS*, pages 128–142. Springer, 2006.

[Kolovos *et al.* 2006b] D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Merging models with the epsilon merging language (eml). In *Proc. MoDELS*, volume 4199 of *Lecture Notes in Computer Science*, pages 215–229. Springer, 2006.

[Kolovos *et al.* 2006c] D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Model comparison: a foundation for model composition and model transformation testing. In *Proc. Workshop on Global Integrated Model Management*, pages 13–20, 2006.

[Kolovos *et al.* 2007a] D.S. Kolovos, R.F. Paige, F.A.C. Polack, and L.M. Rose. Update transformations in the small with the Epsilon Wizard Language. *Journal of Object Technology*, 6(9):53–69, 2007.

[Kolovos *et al.* 2007b] D.S. Kolovos, R.F. Paige, L.M. Rose, and F.A.C Polack. Bridging the Epsilon Wizard Language and the Eclipse Graphical

Modeling Framework. In *Proc. Eclipse Summit*, Ludwigsburg, Germany, 2007.

[Kolovos *et al.* 2008a] D.S. Kolovos, R.F. Paige, and F.A.C Polack. On the evolution of OCL for capturing structural constraints in modelling languages. In *Proc. Workshop on Rigorous Methods for Software Construction and Analysis*, 2008.

[Kolovos *et al.* 2008b] D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Scalability : The holy grail of model driven engineering. In *Proc. Workshop on Challenges in Model Driven Engineering*, 2008.

[Kolovos *et al.* 2008c] D.S. Kolovos, R.F. Paige, and F.A.C Polack. The Epsilon Transformation Language. In *Proc. ICMT*, volume 5063 of *LNCS*, pages 46–60. Springer, 2008.

[Kolovos *et al.* 2009] D.S. Kolovos, R.F. Paige, and L.M. Rose. EuGENia: GMF for mortals. Long talk at Eclipse Summit Europe, October 2009, Ludwigsburg, Germany. Available at: `https://www.eclipsecon.org/submissions/ese2009/view_talk.php?id=979` [Accessed 12 April 2010], 2009.

[Kolovos 2009] D.S. Kolovos. *An Extensible Platform for Specification of Integrated Languages for Model Management*. PhD thesis, University of York, United Kingdom, 2009.

[Kramer 2001] D. Kramer. XEM: XML Evolution Management. Master's thesis, Worcester Polytechnic Institute, MA, USA, 2001.

[Kurtev 2004] I. Kurtev. *Adaptability of Model Transformations*. PhD thesis, University of Twente, Netherlands, 2004.

[Lago *et al.* 2009] P. Lago, H. Muccini, and H. van Vliet. A scoped approach to traceability management. *Systems and Software*, 82(1):168–182, 2009.

[Lämmel & Verhoef 2001] R. Lämmel and C. Verhoef. Semi-automatic grammar recovery. *Software - Practice and Experience*, 31(15):1395–1438, 2001.

[Lämmel 2001] R. Lämmel. Grammar adaptation. In *Proc. Formal Methods for Increasing Software Productivity (FME), International Symposium of Formal Methods Europe*, volume 2021 of *LNCS*, pages 550–570. Springer, 2001.

[Lämmel 2002] R. Lämmel. Towards generic refactoring. In *Proc. ACM SIGPLAN Workshop on Rule-Based Programming*, pages 15–28. ACM, 2002.

[Lara & Guerra 2010] J. de Lara and E. Guerra. Generic meta-modelling with concepts, templates and mixin layers. In *Proc. MoDELS (1)*, volume 6394 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2010.

[Lehman 1969] M.M. Lehman. The programming process. Technical report, IBM Res. Rep. RC 2722, 1969.

[Lerner 2000] B.S. Lerner. A model for compound type changes encountered in schema evolution. *ACM Transactions on Database Systems*, 25(1):83–127, 2000.

[Mäder *et al.* 2008] P. Mäder, O. Gotel, and I. Philippow. Rule-based maintenance of post-requirements traceability relations. In *Proc. IEEE International Requirements Engineering Conference (RE)*, pages 23–32, 2008.

[Martin & Martin 2006] R.C. Martin and M. Martin. *Agile Principles, Patterns, and Practices in C#*. Prentice Hall, Upper Saddle River, NJ, USA, 2006.

[McCarthy 1978] J. McCarthy. History of Lisp. *History of Programming Languages*, 1:217–223, 1978.

[McNeile 2003] A. McNeile. MDA: The vision with the hole? [Accessed 30 June 2008] Available at: `http://www.metamaxim.com/download/documents/MDAv1.pdf`, 2003.

[Mellor & Balcer 2002] S.J. Mellor and M. Balcer. *Executable UML: A Foundation for Model-Driven Architectures*. Addison-Wesley Longman, Boston, Massachusetts, 2002.

[Melnik 2004] S. Melnik. *Generic Model Management: Concepts and Algorithms*. PhD thesis, University of Leipzig, Germany, 2004.

[Méndez *et al.* 2010] D. Méndez, A. Etien, A. Muller, and R. Casallas. Towards transformation migration after metamodel evolution. In *Proc. ME Workshop*, 2010.

[Mens & Demeyer 2007] T. Mens and S. Demeyer. *Software Evolution*. Springer-Verlag, Berlin, 2007.

[Mens & Tourwé 2004] T. Mens and T. Tourwé. A survey of software refactoring. *IEEE Transactions on Software Engineering*, 30(2):126–139, 2004.

[Mens *et al.* 2007] T. Mens, G. Taentzer, and D. Müller. Challenges in model refactoring. In *Proc. Workshop on Object-Oriented Reengineering*, 2007.

[Merriam-Webster 2010] Merriam-Webster. Definition of Nuclear Family. `http://www.merriam-webster.com/dictionary/nuclear%20family`, 2010.

[Moad 1990] J. Moad. Maintaining the competitive edge. *Datamation*, 36(4):61–66, 1990.

[Moha *et al.* 2009] N. Moha, V. Mahé, O. Barais, and J.M. Jézéquel. Generic model refactorings. In *Proc. MoDELS*, volume 5795 of *LNCS*, pages 628–643. Springer, 2009.

[Muller & Hassenforder 2005] P. Muller and M. Hassenforder. HUTN as a Bridge between ModelWare and GrammarWare. In *Proc. Workshop in Software Modelling Engineering*, 2005.

[Nentwich *et al.* 2003] C. Nentwich, W. Emmerich, A. Finkelstein, and E. Ellmer. Flexible consistency checking. *ACM Transactions on Software Engineering and Methodology*, 12(1):28–63, 2003.

[Nguyen *et al.* 2005] T.N. Nguyen, C. Thao, and E.V. Munson. On product versioning for hypertexts. In *Proc. International Workshop on Software Configuration Management (SCM)*, pages 113–132. ACM, 2005.

[Nickel *et al.* 2000] U. Nickel, J. Niere, and A. Zündorf. The FUJABA environment. In *Proc. International Conference on Software Engineering (ICSE)*, pages 742–745, New York, NY, USA, 2000. ACM.

[Northrop 2006] L. Northrop. Ultra-large scale systems: The software challenge of the future. Technical report, Carnegie Mellon, June 2006.

[Oldevik *et al.* 2005] J. Oldevik, T. Neple, R. Grønmo, J.Ø. Aagedal, and A. Berre. Toward standardised model to text transformations. In *Proc. ECMDA-FA*, volume 3748 of *LNCS*, pages 239–253. Springer, 2005.

[Olsen & Oldevik 2007] G.K. Olsen and J. Oldevik. Scenarios of traceability in model to text transformations. In *Proc. ECMDA-FA*, volume 4530 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 2007.

[OMG 2001] OMG. Unified Modelling Language 1.4 Specification [online]. [Accessed 15 September 2008] Available at: `http://www.omg.org/spec/UML/1.4/`, 2001.

[OMG 2004] OMG. Human-Usable Textual Notation 1.0 Specification [online]. [Accessed 30 June 2008] Available at: `http://www.omg.org/technology/documents/formal/hutn.htm`, 2004.

[OMG 2005] OMG. MOF QVT Final Adopted Specication [online]. [Accessed 22 July 2009] Available at: `www.omg.org/docs/ptc/05-11-01.pdf`, 2005.

[OMG 2006] OMG. Object Constraint Language 2.0 Specification [online]. [Accessed 30 June 2008] Available at: `http://www.omg.org/technology/documents/formal/ocl.htm`, 2006.

[OMG 2007a] OMG. Unified Modelling Language 2.1.2 Specification [online]. [Accessed 30 June 2008] Available at: http://www.omg.org/spec/UML/2.1.2/, 2007.

[OMG 2007b] OMG. Unified Modelling Language 2.2 Specification [online]. [Accessed 5 March 2010] Available at: http://www.omg.org/spec/UML/2.2/, 2007.

[OMG 2007c] OMG. XML Metadata Interchange 2.1.1 Specification [online]. [Accessed 30 June 2008] Available at: http://www.omg.org/technology/documents/formal/xmi.htm, 2007.

[OMG 2008a] OMG. Meta-Object Facility [online]. [Accessed 30 June 2008] Available at: http://www.omg.org/mof, 2008.

[OMG 2008b] OMG. Model Driven Architecture [online]. [Accessed 30 June 2008] Available at: http://www.omg.org/mda/, 2008.

[OMG 2008c] OMG. Object Management Group home page [online]. [Accessed 30 June 2008] Available at: http://www.omg.org, 2008.

[Opdyke 1992] W.F. Opdyke. *Refactoring Object-Oriented Frameworks*. PhD thesis, University of Illinois at Urbana-Champaign, IL, USA, 1992.

[openArchitectureWare 2007] openArchitectureWare. openArchitectureWare Project Website [online]. [Accessed 30 June 2008] Available at: http://www.eclipse.org/gmt/oaw/, 2007.

[openArchitectureWare 2008] openArchitectureWare. XPand Language Reference [online]. [Accessed 18 August 2010] Available at: http://wiki.eclipse.org/AMMA, 2008.

[Paige *et al.* 2007] R.F. Paige, P.J. Brooke, and J.S. Ostroff. Metamodel-based model conformance and multiview consistency checking. *ACM Transactions on Software Engineering and Methodology*, 16(3), 2007.

[Paige *et al.* 2009] R.F. Paige, L.M. Rose, X. Ge, D.S. Kolovos, and P.J. Brooke. FPTC: Automated safety analysis for domain-specific languages. In *MoDELS Workshops and Symposia*, volume 5421 of *Lecture Notes in Computer Science*, pages 229–242. Springer, 2009.

[Parr 2007] T. Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Programmers, 2007.

[Patrascoiu & Rodgers 2004] O. Patrascoiu and P. Rodgers. Embedding OCL expressions in YATL. In *Proc. OCL and Model-Driven Engineering Workshop*, 2004.

[Pilgrim *et al.* 2008] J. von Pilgrim, B. Vanhooff, I. Schulz-Gerlach, and Y. Berbers. Constructing and visualizing transformation chains. In *Proc. European Conference on the Model Driven Architecture – Foundations and Applications*, volume 5095 of *LNCS*, pages 17–32. Springer, 2008.

[Pizka & Jürgens 2007] M. Pizka and E. Jürgens. Automating language evolution. In *Proc. Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 305–315. IEEE Computer Society, 2007.

[Porres 2003] I. Porres. Model refactorings as rule-based update transformations. In *Proc. UML*, volume 2863 of *LNCS*, pages 159–174. Springer, 2003.

[RAE & BCS 2004] The RAE and The BCS. The challenges of complex IT projects. Technical report, The Royal Academy of Engineering, April 2004.

[Ramil & Lehman 2000] J.F. Ramil and M.M. Lehman. Cost estimation and evolvability monitoring for software evolution processes. In *Proc. Workshop on Empirical Studies of Software Maintenance*, 2000.

[Ráth *et al.* 2008] I. Ráth, G. Bergmann, A. Ökrös, and D. Varró. Live model transformations driven by incremental pattern matching. In *Proc. ICMT*, volume 5063 of *LNCS*, pages 107–121. Springer, 2008.

[Rising 2001] L. Rising, editor. *Design patterns in communications software.* Cambridge University Press, 2001.

[Rose *et al.* 2008a] L.M. Rose, R.F. Paige, D.S. Kolovos, and F.A.C. Polack. Constructing models with the Human-Usable Textual Notation. In *Proc. International Conference on Model Driven Engineering Languages and Systems*, volume 5301 of *LNCS*, pages 249–263. Springer, 2008.

[Rose *et al.* 2008b] L.M. Rose, R.F. Paige, D.S. Kolovos, and F.A.C. Polack. The Epsilon Generation Language. In *Proc. European Conference on Model Driven Architecture – Foundations and Applications*, volume 5095 of *LNCS*, pages 1–16. Springer, 2008.

[Rose *et al.* 2009a] L.M. Rose, D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Enhanced automation for managing model and metamodel inconsistency. In *Proc. ASE*, pages 545–549. ACM Press, 2009.

[Rose *et al.* 2009b] L.M. Rose, R.F. Paige, D.S. Kolovos, and F.A.C.s Polack. An analysis of approaches to model migration. In *Proc. Joint MoDSE-MCCM Workshop*, 2009.

[Rose *et al.* 2010a] L.M. Rose, A. Etien, D. Méndez, D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Comparing model-metamodel and transformation-metamodel co-evolution. In *Proc. ME Workshop*, 2010.

[Rose *et al.* 2010b] L.M. Rose, M. Herrmannsdoerfer, J.R. Williams, D.S. Kolovos, K. Garcés, R.F. Paige, and F.A.C. Polack. A comparison of model migration tools. In *Proc. MoDELS*, volume TBC of *Lecture Notes in Computer Science*, page TBC. Springer, 2010.

[Rose *et al.* 2010c] L.M. Rose, D.S. Kolovos, N. Drivalos, J.R. Williams, R.F. Paige, F.A.C. Polack, and K.J. Fernandes. Concordance: An efficient framework for managing model integrity. In *Proc. European Conference on Modelling Foundations and Applications*, volume 6138 of *LNCS*, pages 62–73. Springer, 2010.

[Rose *et al.* 2010d] L.M. Rose, D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Migrating activity diagrams with Epsilon Flock. In *Proc. TTC*, 2010.

[Rose *et al.* 2010e] L.M. Rose, D.S. Kolovos, R.F. Paige, and F.A.C. Polack. Model migration case. In *Proc. TTC*, 2010.

[Rose *et al.* 2010f] L.M. Rose, D.S. Kolovos, R.F. Paige, and F.A.C Polack. Model migration with Epsilon Flock. In *Proc. ICMT*, volume 6142 of *Lecture Notes in Computer Science*, pages 184–198. Springer, 2010.

[Selic 2003] B. Selic. The pragmatics of Model-Driven Development. *IEEE Software*, 20(5):19–25, 2003.

[Selic 2005] B. Selic. Whats new in UML 2.0? *IBM Rational software*, 2005.

[Sendall & Kozaczynski 2003] S. Sendall and W. Kozaczynski. Model transformation: The heart and soul of model-driven software development. *IEEE Software*, 20:42–45, 2003.

[Sjøberg 1993] D.I.K. Sjøberg. Quantifying schema evolution. *Information & Software Technology*, 35(1):35–44, 1993.

[Sommerville 2006] I. Sommerville. *Software Engineering*. Addison-Wesley, Boston, Massachusetts, 9th edition, 2006.

[Sprinkle & Karsai 2004] J. Sprinkle and G. Karsai. A domain-specific visual language for domain model evolution. *Journal of Visual Languages and Computing*, 15(3-4):291–307, 2004.

[Sprinkle 2003] J. Sprinkle. *Metamodel Driven Model Migration*. PhD thesis, Vanderbilt University, TN, USA, 2003.

[Stahl *et al.* 2006] T. Stahl, M. Voelter, and K. Czarnecki. *Model-Driven Software Development: Technology, Engineering, Management.* Wiley, Chichester, 2006.

[Starfield *et al.* 1990] M. Starfield, K.A. Smith, and A.L. Bleloch. *How to model it: Problem Solving for the Computer Age.* McGraw-Hill Inc., New York, 1990.

[Steinberg *et al.* 2008] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks. *EMF: Eclipse Modeling Framework.* Addison-Wesley Professional, Boston, Massachusetts, 2008.

[Su *et al.* 2001] H. Su, D. Kramer, L. Chen, K.T. Claypool, and E.A. Rundensteiner. XEM: Managing the evolution of XML documents. In *Proc. Workshop on Research Issues in Data Engineering*, pages 103–110, 2001.

[Tisi *et al.* 2009] M. Tisi, F. Jouault, P. Fraternali, S. Ceri, and J. Bézivin. On the use of higher-order model transformations. In *Proc. ECMDA-FA*, volume 5562 of *Lecture Notes in Computer Science*, pages 18–33. Springer, 2009.

[Tratt 2008] L. Tratt. A change propagating model transformation language. *Journal of Object Technology*, 7(3):107–124, 2008.

[Varró & Balogh 2007] D. Varró and A. Balogh. The model transformation language of the VIATRA2 framework. *Science of Computer Programming*, 68(3):187–207, 2007.

[Vries & Roddick 2004] D. de Vries and J.F. Roddick. Facilitating database attribute domain evolution using meso-data. In *Proc. Workshop on Evolution and Change in Data Management*, pages 429–440, 2004.

[W3C 2007a] W3C. W3C XML Schema 1.1 Specification [online]. [Accessed 30 June 2008] Available at: `http://www.w3.org/XML/Schema`, 2007.

[W3C 2007b] W3C. World Wide Web Consortium [online]. [Accessed 30 June 2008] Available at: `http://www.w3.org/`, 2007.

[Wachsmuth 2007] G. Wachsmuth. Metamodel adaptation and model co-adaptation. In *Proc. ECOOP*, volume 4609 of *LNCS*, pages 600–624. Springer, 2007.

[Wallace 2005] M. Wallace. Modular architectural representation and analysis of fault propagation and transformation. *Electronic Notes in Theoretical Computer Science*, 141(3):53–71, 2005.

[Ward 1994] M.P. Ward. Language-oriented programming. *Software — Concepts and Tools*, 15(4):147–161, 1994.

[Watson 2008] A. Watson. A brief history of MDA. *Upgrade*, 9(2), 2008.

[Welch & Barnes 2005] P.H. Welch and F.R.M. Barnes. Communicating mobile processes. In *Proc. Symposium on the Occasion of 25 Years of Communicating Sequential Processes (CSP)*, volume 3525 of *LNCS*, pages 175–210. Springer, 2005.

[Winkler & Pilgrim 2009] S. Winkler and J. von Pilgrim. A survey of traceability in requirements engineering and model-driven development. *Software and Systems Modeling*, December 2009.