

GECO: Automatic Generator-Composition for (Aspect-oriented) DSLs

Doctoral Symposium

Reiner Jung

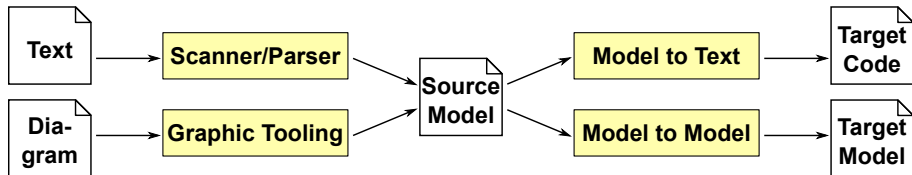
Christian-Albrechts-Universität zu Kiel
Institut für Informatik

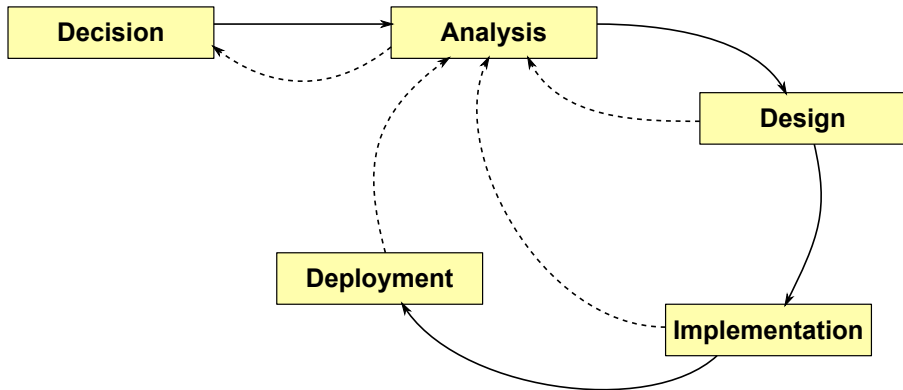
26.02.2014



Domain-specific Language (DSL)

- ▶ Created for a specific domain or aspect of a software system
- ▶ Properties: formal, informal, executable etc.
- ▶ Text-based or graphical languages
 - ▶ Text-based: grammars and meta-models
 - ▶ Graphical: visualization pattern and meta-models





[Mernik et al., 2005]

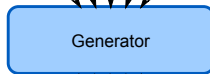
Automatic or semi-automatic tasks

- ▶ Meta-model generation
- ▶ Editor generation

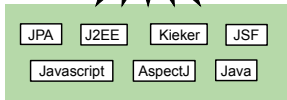
Manual tasks

- ▶ Grammar specification
- ▶ Validator construction
- ▶ Generator construction

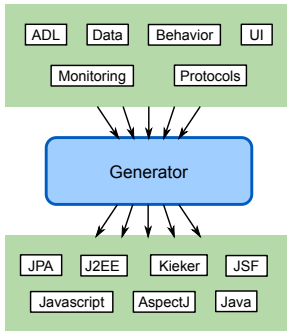
Source Meta-Models /
DSLs



Target Meta-Models /
Technologies &
General Purpose
Languages (GPL)

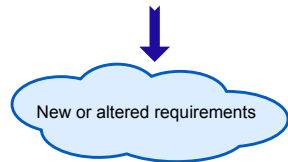


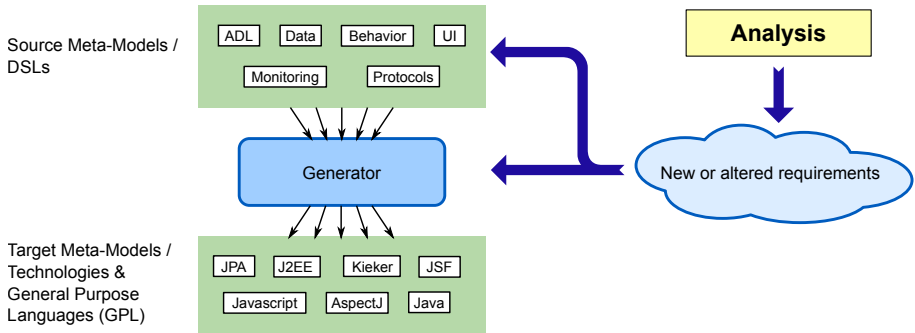
Source Meta-Models /
DSLs



Target Meta-Models /
Technologies &
General Purpose
Languages (GPL)

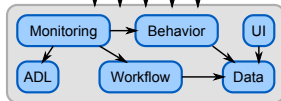
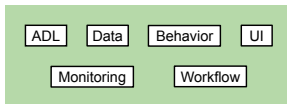
Analysis



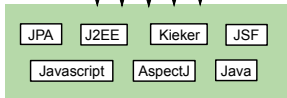


Meta-model changes result in generator code degradation

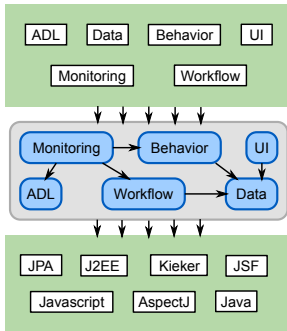
Source Meta-Models /
DSLs



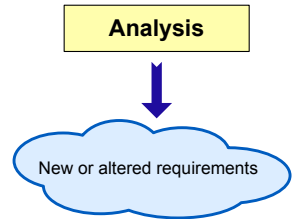
Target Meta-Models /
Technologies &
General Purpose
Languages (GPL)



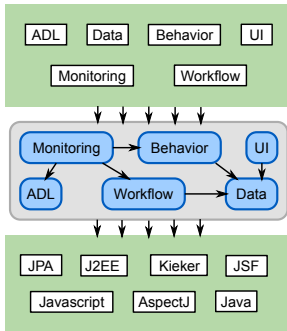
Source Meta-Models /
DSLs



Target Meta-Models /
Technologies &
General Purpose
Languages (GPL)



Source Meta-Models /
DSLs

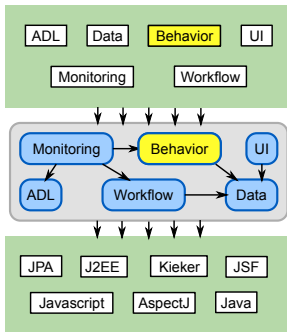


Target Meta-Models /
Technologies &
General Purpose
Languages (GPL)

Analysis

New or altered requirements

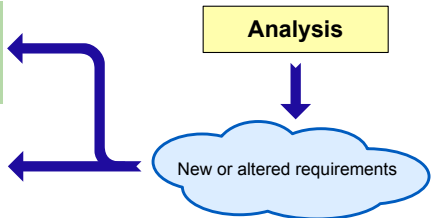
Source Meta-Models /
DSLs

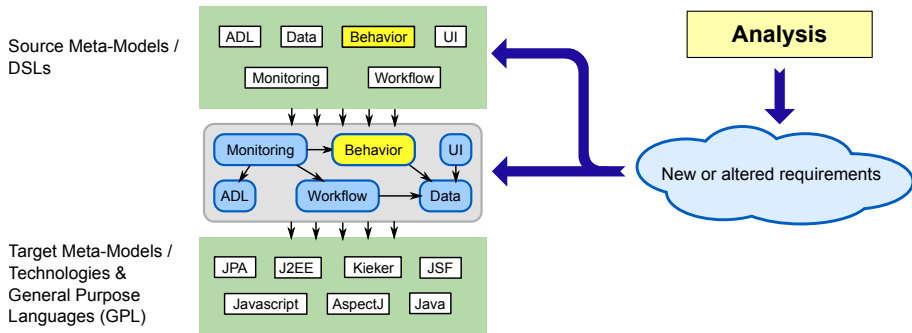


Target Meta-Models /
Technologies &
General Purpose
Languages (GPL)

Analysis

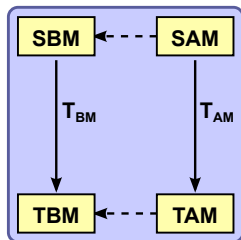
New or altered requirements





Key problem How to combine partial generators?

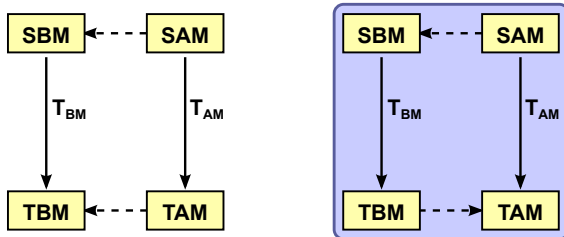
SBM	Source Base Model	TBM	Target Base Model	\xleftarrow{T}	Transformation
SAM	Source Aspect Model	TAM	Target Aspect Model	$\xleftarrow{---}$	References
		TM	Target Model		



SBM Source Base Model
SAM Source Aspect Model

TBM Target Base Model
TAM Target Aspect Model
TM Target Model

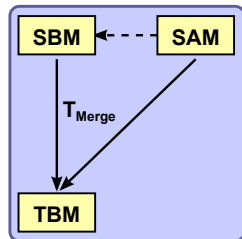
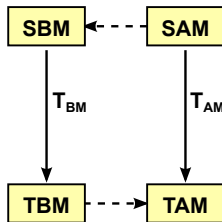
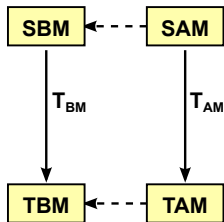
←^T Transformation
←- - References



SBM Source Base Model
SAM Source Aspect Model

TBM Target Base Model
TAM Target Aspect Model
TM Target Model

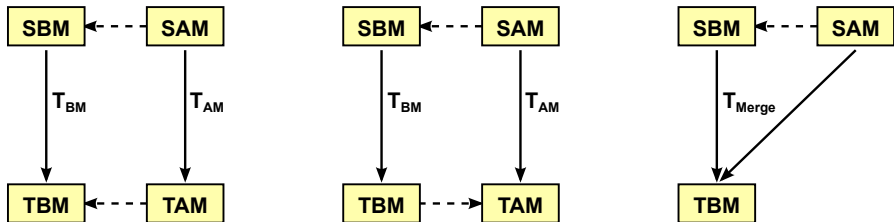
\xrightarrow{T} Transformation
 $\xleftarrow{- - -}$ References



SBM Source Base Model
SAM Source Aspect Model

TBM Target Base Model
TAM Target Aspect Model
TM Target Model

\xrightarrow{T} Transformation
 $\leftarrow \text{---}$ References



SBM Source Base Model
SAM Source Aspect Model

TBM Target Base Model
TAM Target Aspect Model
TM Target Model

\xrightarrow{T} Transformation
 \dashleftarrow References

Approach 1 Weaving models

- ▶ Base and aspect model generators produce target models
- ▶ Reference resolver for join points (source to target level)
- ▶ Weaver combines target aspect and target base model
- ▶ Various target level technologies (AspectJ, Kermeta, . . .)

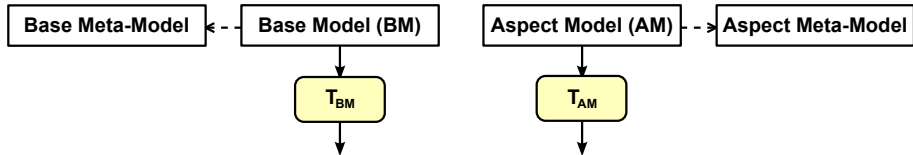
Approach 2 Weaving partial generators

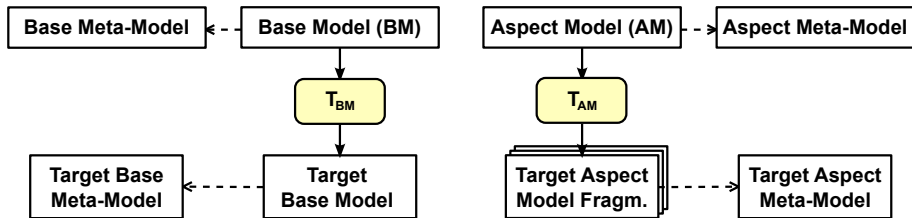
- ▶ Base and aspect generator are woven
- ▶ Integration based on source level meta-model references
- ▶ Requires strict coding guidelines and insertion pattern
- ▶ Can be based on higher order transformations [Tisi et al., 2009]

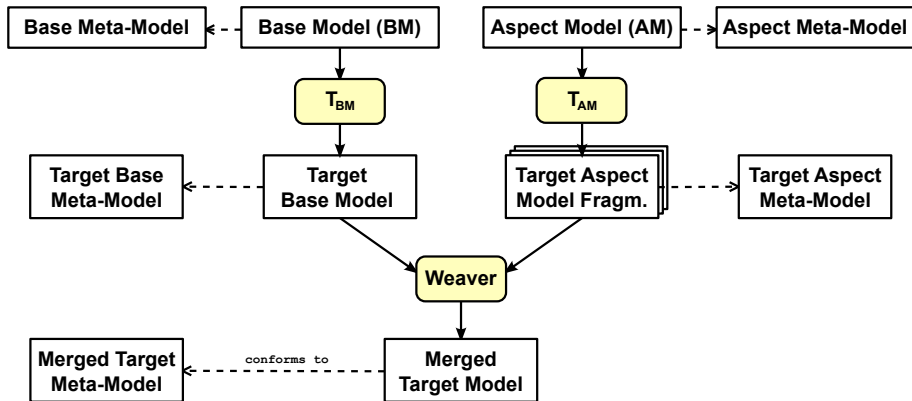
Base Model (BM)

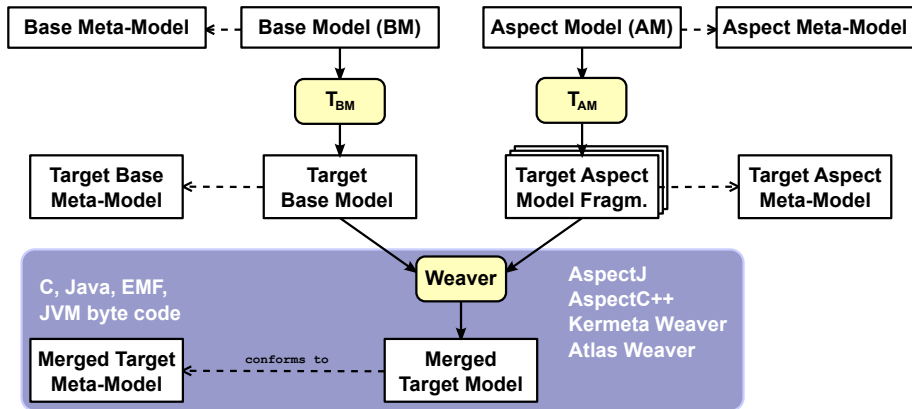
Aspect Model (AM)

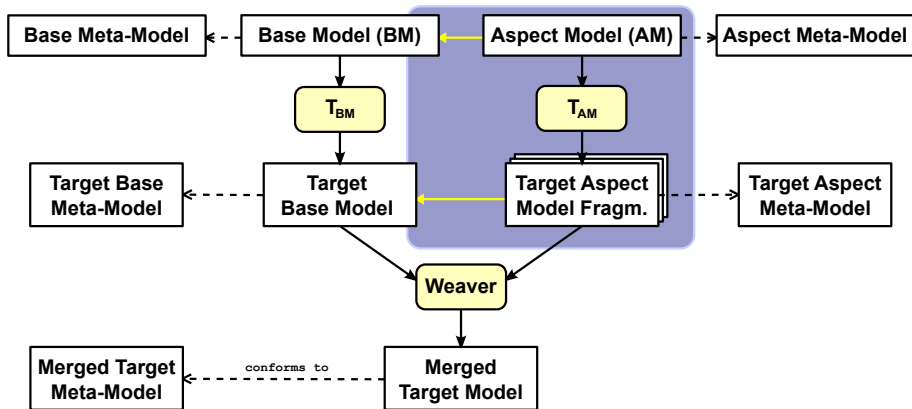


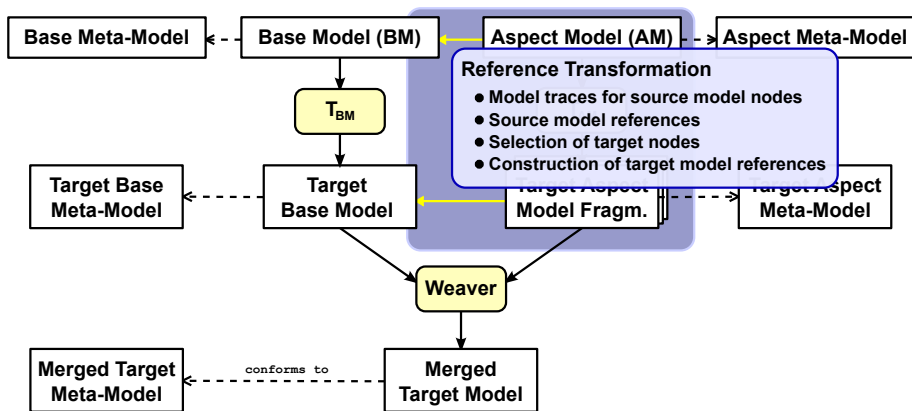












Generator for CoCoME [Rausch et al., 2011]

- ▶ Domain: Common Component Modeling Example (CoCoME)
- ▶ Users: DFG-SPP *Design for Future* projects
- ▶ Target platform: Cloud, J2EE, JSF, Java, AspectJ

Design For
FUTURE

iObserve



Generator for MENGES DSL [Goerigk et al., 2012]

- ▶ Domain: Railway control centers (RCC)
- ▶ Users: Developers and engineers of RCC
- ▶ Target platform: PLCopen/XML, C

MENGES
KoSSE

SCHEIDT&BACHMANN 

b+m
business IT management

- Goal A Technique and method evaluation
- Goal B Approach evaluation in an enterprise scenario
- Goal C Approach evaluation in a legacy scenario with multiple target platforms

Objectives

- ▶ Feasibility of different model-element reference concepts
- ▶ Feasibility of model/code weaving approach
- ▶ Practicability of model-element reference concepts
 - ▶ With researchers of iObserve projects and students

Scenario

- ▶ Partial CoCoME model, based on the Palladio Component Model (PCM) [Becker et al., 2009]
- ▶ Instrumentation aspect language (IAL) [Jung et al., 2013a]
- ▶ Combination of IAL & ProtoCom [Giacinto and Lehrig, 2013]

Objectives

- ▶ Integration test of method and tooling
- ▶ Practicability of approach and tooling (J2EE domain)

Scenario

- ▶ Complete CoCoME model based on PCM and additional DSLs
- ▶ Adaptation of generators driven by CoCoME modification scenarios

Collaboration

- ▶ University Stuttgart
- ▶ UDE, KIT, CAU
- ▶ Other CoCoME collaborators



iObserve

Design For
FUTURE

Objectives

- ▶ Feasible for different target model/language domains (text, XML)
- ▶ Practicability of approach and tooling (RCC domain)
- ▶ Comparison with existing generator \Rightarrow cost benefit

Scenario

- ▶ Legacy meta-model and type-system
- ▶ Creation of generators for different target languages

Collaboration & Expert Interviews

- ▶ b+m Informatik, Stefan Zeug
- ▶ Scheidt & Bachmann, Hauke Fuhrmann



Published

- ▶ Type-Systems for DSLs [Jung et al., 2013b]
- ▶ Data type language (DTL) [Jung, 2013]
- ▶ Instrumentation aspect language [Jung et al., 2013a]

Unpublished

- ▶ Survey on model join point notations
- ▶ Designing extensible meta-models (addressing PCM issues)

WP1 Literature and Technology Research

- Model weaver & graph transformation
- Compiler construction

WP2 Communication

- More publications and presentations

WP3 Approach

WP4 Solution Design and Implementation

WP5 Evaluation

Aspect Oriented-Modeling (AOM)

- ▶ Formal Design Analysis Framework (FDAF) [Bennett et al., 2010]
 - ▶ Supports only UML class diagrams
 - ▶ Stub generation Java and AspectJ
- ▶ Theme/UML [Clarke and Baniassad, 2005]
 - ▶ Supports UML structural and behavior models
 - ▶ No code generation
 - ▶ Themes as aspects
- ▶ Reusable Aspect Models (RAM) [Klein and Kienzle, 2007]
 - ▶ Supports UML class and activity models
 - ▶ Weaves in modeling domain
 - ▶ Code generation for Java and AspectJ

Aspect Oriented-Modeling (AOM)

- ▶ Formal Design Analysis Framework (FDAF) [Bennett et al., 2010]
 - ▶ Supports only UML class diagrams
 - ▶ Stub generation Java and AspectJ
- ▶ Theme/UML [Clarke and Baniassad, 2005]
 - ▶ Supports UML structural and behavior models
 - ▶ No code generation
 - ▶ Themes as aspects
- ▶ Reusable Aspect Models (RAM) [Klein and Kienzle, 2007]
 - ▶ Supports UML class and activity models
 - ▶ Weaves in modeling domain
 - ▶ Code generation for Java and AspectJ

Code Generation

- ▶ No work on iterative/agile development [Mehmood and Jawawi, 2013]

Summary

- ▶ DSLs are developed iteratively
- ▶ Generators can depend on multiple DSLs
 - ▶ Resulting in complex generators
- ▶ Iterations of each DSL cause
 - ▶ Generator architecture degradation

Goal Provide an approach and tooling to improve generator

- ▶ construction
- ▶ re-use

Next step Realizing the first evaluation scenario

Release date July/August 2015

- S. Becker, H. Koziol, and R. Reussner. The palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3 – 22, 2009. ISSN 0164-1212. doi: <http://dx.doi.org/10.1016/j.jss.2008.03.066>. URL <http://www.sciencedirect.com/science/article/pii/S0164121208001015>. Special Issue: Software Performance - Modeling and Analysis.
- J. Bennett, K. Cooper, and L. Dai. Aspect-oriented model-driven skeleton code generation: A graph-based transformation approach. *Science of Computer Programming*, 75(8):689 – 725, 2010. ISSN 0167-6423. doi: <http://dx.doi.org/10.1016/j.scico.2009.05.005>. URL <http://www.sciencedirect.com/science/article/pii/S0167642309000914>. Designing high quality system/software architectures.
- S. Clarke and E. Baniassad. *Aspect-Oriented Analysis and Design*. Addison-Wesley Professional, 2005. ISBN 0321246748.
- D. Giacinto and S. Lebrig. Towards integrating java ee into protocom. In S. Becker, W. Hasselbring, A. van Hoorn, and R. Reussner, editors, *KPDAAYS*, volume 1083 of *CEUR Workshop Proceedings*, pages 69–78. CEUR-WS.org, 2013. URL <http://dblp.uni-trier.de/db/conf/kpdays/kpdays2013.html#GiacintoL13>.

- W. Goerigk, R. von Hanxleden, W. Hasselbring, G. Hennings, R. Jung, H. Neustock, H. Schaefer, C. Schneider, E. Schultz, T. Stahl, S. Weik, and S. Zeug. Entwurf einer domänenspezifischen sprache für elektronische stellwerke. In *Software Engineering 2012: Fachtagung des GI-Fachbereichs Softwaretechnik, 27. Februar - 2. März 2012 in Berlin*, volume 198 of *Lecture Notes in Informatics*, pages 119–130. Gesellschaft für Informatik e.V., 2012. URL <http://eprints.uni-kiel.de/14420/>.
- R. Jung. Data type language, November 2013. URL http://www.dfg-spp1593.de/wiki/index.php/Model-driven_development_of_CoCoME.
- R. Jung, R. Heinrich, and E. Schmieders. Model-driven instrumentation with kieker and palladio to forecast dynamic applications. *Symposium on Software Performance: Joint Kieker/Palladio Days 2013*, 1083:99–108, November 2013a. URL <http://eprints.uni-kiel.de/22655/>.
- R. Jung, C. Schneider, and W. Hasselbring. Type systems for domain-specific languages. In S. Wagner and H. Lichter, editors, *Software Engineering 2013 Workshopband*, volume 215 of *Lecture Notes in Informatics*, pages 139–154, Bonn, February 2013b. Gesellschaft für Informatik e.V. URL <http://eprints.uni-kiel.de/20641/>.
- J. Klein and J. Kienzle. Reusable aspect models. In *11th Workshop on Aspect-Oriented Modeling, AOM at Models'07*,, 2007.

- A. Mehmood and D. N. Jawawi. Aspect-oriented model-driven code generation: A systematic mapping study. *Information and Software Technology*, 55(2):395 – 411, 2013. ISSN 0950-5849. doi: <http://dx.doi.org/10.1016/j.infsof.2012.09.003>. URL <http://www.sciencedirect.com/science/article/pii/S0950584912001863>. Special Section: Component-Based Software Engineering (CBSE), 2011.
- M. Mernik, J. Heering, and A. M. Sloane. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, Dec. 2005. ISSN 0360-0300. doi: 10.1145/1118890.1118892. URL <http://doi.acm.org/10.1145/1118890.1118892>.
- A. Rausch, R. Reussner, R. Mirandola, and F. Plasil, editors. *The Common Component Modelling Example (CoCoME)*, volume 5153 of *Lecture Notes in Computer Science*. Springer Verlag Berlin Heidelberg, 2011.
- M. Tisi, F. Jouault, P. Fraternali, S. Ceri, and J. Bézivin. On the use of higher-order model transformations. In R. Paige, A. Hartman, and A. Rensink, editors, *Model Driven Architecture - Foundations and Applications*, volume 5562 of *Lecture Notes in Computer Science*, pages 18–33. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02673-7. doi: 10.1007/978-3-642-02674-4_3. URL http://dx.doi.org/10.1007/978-3-642-02674-4_3.