



Universiteit
Leiden
The Netherlands

Abstract delta modeling : software product lines and beyond

Helvensteijn, M.

Citation

Helvensteijn, M. (2014, November 12). *Abstract delta modeling : software product lines and beyond*. *IPA Dissertation Series*. Retrieved from <https://hdl.handle.net/1887/29661>

Version: Not Applicable (or Unknown)

License: [Leiden University Non-exclusive license](#)

Downloaded from: <https://hdl.handle.net/1887/29661>

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

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/29661> holds various files of this Leiden University dissertation

Author: Helvensteijn, Michiel

Title: Abstract delta modeling : software product lines and beyond

Issue Date: 2014-11-12

Curriculum Vitæ

- 1997–1998 • MBO Praktische Informatiekunde 1 at Stichting Facta
- 1999–2000 • MBO Algemene Ondernemers Vaardigheden at the NHA
- 1998–2004 • Voorbereidend Wetenschappelijk Onderwijs at Alfrink College Zoetermeer
- 2004–2007 • Bachelor of Science in Informatica at Leiden University (*cum laude*)
- 2007–2009 • Master of Science in Computer Science at Leiden University (*cum laude*)
- 2009–2014 ◦ PhD in Computer Science at the CWI and Leiden University

Index

A

ABS, *see* Abstract Behavioral Specification
absorption, **23**
Abstract Behavioral Specification, **155**
Abstract Delta Modeling, **6**
accessability relation, **25**
ADM, *see* Abstract Delta Modeling
advice, **5**
AHEAD, **65**
algebra, **24**
 Boolean, **24**
 quotient, **24**
 relation, **24**
algebraic signature, **24**
algebraic structure, **24**
annotated delta model, **103**
annotated delta model closure, **115**
annotative variability, **6**
antisymmetry, **21**
AOP, *see* Aspect Oriented Programming
application condition, **102**
 joint, **104**
application function, **102**
application order, **73**
arity, **23**
around advice, **83**
aspect, **5**
Aspect Oriented Programming, **5**
associativity, **23**
asymmetry, **21**
automated product derivation, **97**
automated profile management, **168**

B

bijectivity, **21**
Boolean algebra, **24**

C

cardinality, **18**
carrier set, **24**
Cartesian product, **19**
 n -ary, **19**
choice axiom, **138**
CIDE, **94**
class, **31**
closure
 annotated delta model, **115**
 delta model, **91**
 reflexive transitive, **21**
 transitive, **21**
coarse-grained, **40**
code view, **206**
codomain, **22**
commutativity, **23**
composition axiom, **138**
compositional variability, **6**
concatenation, sequence, **19**
concern, *see* feature
 cross-cutting, **4**
conclusion, **21**
configuration, **27**
 difference-based, **178**
 minimal, **175**
 stable, **179**
 stuck, **28**
 unstable, **179**
configuration space, **27**
conflict, *see* delta conflict
conflict relation, **76**
conflict resolution job, **154**

conflict resolution relation, **78**
 conflict resolving delta, **78**
 conjunctive delta Kripke frame, **137**
 conjunctive delta model evaluation,
 88
 conjunctive semantics, **88**
 consensus axiom, **138**
 consistent conflict resolution, **81**
 constant, **23**
 constraint, **170**
 constructivism, **52**
 context-aware program, **194**
 core product, **103**
 correctness
 delta, **45**
 dynamic product line, **177**
 product line, **108**
 cost domain, **189**
 cross-cutting concern, **4**
 current state, **180**

D

decision modeling, **116**
 deeply annotated delta model, **114**
 deeply annotated delta model eval-
 uation, **115**
 delta, **36**
 conflict resolving, **78**
 deterministic, **44**
 empty, **51**
 feature implementation, **153**
 feature interaction, **153**
 fine-grained software, **84**
 fully defined, **44**
 invalid, *see* delta, undefined
 $\text{\LaTeX}$, **122**
 neutral, **50**
 non-deterministic, **44**
 parametric, **111**
 parametric software, **112**
 partially defined, **44**
 semantic, **37**
 semantic equivalence, **48**
 semantic refinement, **47**
 simple, **90**
 software, **39**
 software class, **39**
 software package, **39**
 statement, **84**
 undefined, **44**

delta application, **37**
 delta choice, **51**
 delta commutativity, **50**
 delta composition, **50**
 delta conflict, **76**
 hard, **79**
 three-way, **110**
 delta consensus, **51**
 delta contract, **140**
 delta contract provability, **141**
 delta converse, **53**
 delta correctness, **45**
 delta derivation, **46**
 delta diagram, **73**
 delta evaluation, **37**
 parametric software, **112**
 delta Kripke frame, **137**
 delta Kripke model, **137**
 delta logic, **138**
 delta model, **73**
 annotated, **103**
 axiom $\Delta\cap$, **138**
 axiom $\Delta\cup$, **138**
 deeply annotated, **114**
 flat, **90**
 local, **151**
 nested, **90**
 nesting, **90**
 parametric, **111**
 selected, **103**
 unambiguous, **79**
 delta model closure, **91**
 delta model evaluation, **75**
 deeply annotated, **115**
 parametric, **111**
 delta modeling, **6**
 Delta Modeling Workflow, **149**
 delta oriented programming, **71**
 delta postcondition, **140**
 delta precondition, **140**
 delta specification, **45**
 delta-aware operation, **122**
 deltoid, **37**
 fine-grained software, **85**
 fully expressive, **56**
 functional, **59**
 $\text{\LaTeX}$, **120**
 maximally expressive, **57**
 parametric, **111**
 parametric software, **112**

- partially functional, **59**
- profile, **171**
- quark, **63**
- quotient, **38**
- relational, **58**
- software, **41**
- stone carving, **57**
- deltoid homomorphism, **58**
- deltoid refinement, **58**
- derivation, **74**
 - nesting-aware, **91**
 - unique, **75**
- derivation function, **74**
- derivative module, **78**
- derived delta operator, **181**
- derived profile delta operator, **181**
- desired feature interaction, **107**
- deterministic delta, **44**
- device, **169**
- difference-based configuration, **178**
- direct path, **183**
- discreteness, **21**
- disjunctive delta Kripke frame, **137**
- disjunctive delta model evaluation, **88**
- disjunctive semantics, **88**
- distribution, **23**
- DMW, *see* Delta Modeling Workflow
- DMW transition relation, **215**
- domain, **22**
- DPL Mealy machine, **182**
- dynamic product line, **167**
- dynamic product line correctness, **177**
- dynamic product line implementation, **175**

E

- Editor product line, **7**
- effect specification, **171**
- element, **18**
 - absorbing, **23**
 - identity, **23**
 - maximal, **22**
 - minimal, **22**
- empty delta, **51**
- empty delta axiom, **138**
- empty introduction, **61**
- empty quark, **62**

- empty set, **19**
- environment, **169**
- environmental feature configuration, **175**
- environmental transition, **176**
- equivalence class, **23**
- equivalence relation, **22**
- expansion law, **139**
- extension, **23**

F

- FAS, *see* Fredhopper Access Server
- feature, **4**, **99**
- feature combination order, **155**
- feature configuration, **100**
 - environmental, **175**
 - local, **178**
- feature configuration equivalence, **190**
- feature diagram, **100**
- feature implementation delta, **153**
- feature implementation job, **153**
- feature initialization problem, **82**
- feature interaction, **75**
- feature interaction delta, **153**
- feature locality, *see* feature modularity
- feature model, **100**
- feature modularity, **34**
- feature module, **6**
- feature optionality problem, **70**
- Feature Oriented Programming, **64**
- Feature Oriented Software Development, **5**
- feature structure tree, **60**
- Feature-oriented Domain Analysis, **99**
- FgSD, *see* fine-grained software deltoid
- field, **31**
- fine-grained, **40**
- fine-grained software delta, **84**
- fine-grained software deltoid, **85**
- flat delta model, **90**
- FODA, *see* Feature-oriented Domain Analysis
- forcing relation, **26**
- FOSD, *see* Feature Oriented Software Development
- Fredhopper Access Server, **161**

Fredhopper Access Server product line, **161**
 FST, *see* feature structure tree
 full difference transition function, **183**
 full quark, **62**
 full-definedness, **21**
 fully defined delta, **44**
 fully expressive deltoid, **56**
 function, **22**
 partial, **22**
 updated, **22**
 functional deltoid, **59**
 functionality, **21**

G

generalization, **26**
 global consequence, **26**
 global quark, **62**
 global unambiguity, **105**
 glue code, **78**
 granularity, **40**

H

hard delta conflict, **79**
 Heyting algebra, **205**
 hybrid language, **207**

I

idempotency, **23**
 identifier, **31**
 identity, **23**
 identity modification, **61**
 identity relation, **20**
 image, **20**
 inclusive order, **22**
 incremental application, **37**
 inference rule, **21**
 infinite transition path, **28**
 initial product, **57**
 initial state, **212**
 initialization block, **156**
 injectivity, **21**
 integers, **19**
 introduction, **61**
 introduction equivalence, **61**
 introduction sum, **61**
 invalid delta, *see* undefined delta
 invalid software delta, **43**
 invasive composition, **6, 40**

irreflexivity, **21**

J

job, **153**
 conflict resolution, **154**
 feature implementation, **153**
 job status map, **211**
 joint application condition, **104**

K

Kripke frame, **25**
 delta, **137**
 Kripke model, **25**
 delta, **137**

L

L^AT_EX, **119**
 L^AT_EX deltoid, **120**
 lattice, **52**
 lifter, **78**
 linear extension, **23**
 local consequence, **26**
 local consistency, **181**
 local delta model, **151**
 local feature configuration, **178**
 local quark, **62**
 local transition, **176**

M

magma, **62**
 manual profile, **174**
 mapping, **22**
 mass customization, **4**
 maximal element, **22**
 maximally expressive deltoid, **57**
 Mealy machine, **28**
 DPL, **182**
 Mealy machine transition relation, **180**
 method, **31**
 minimal element, **22**
 minimal transition function, **187**
 mixin, **66**
 modal label, **25**
 modality, **25**
 modification, **61**
 identity, **61**
 modification application, **61**
 modification product, **61**
 modus ponens, **26**

monkey-patching, **119**
 monoid, **24**
 profile delta, **171**
 multimodal language, **25**

N

naive set theory, **18**
 n -ary Cartesian product, **19**
 n -ary operation, **23**
 n -ary relation, **20**
 natural numbers, **19**
 positive, **19**
 nested delta model, **90**
 nested product line, **114**
 nested product line implementation, **115**
 nesting delta model, **90**
 nesting-aware derivation, **91**
 neutral delta, **50**
 neutral delta axiom, **138**
 nominal, **207**
 non-deterministic delta, **44**
 non-interference, **211**
 normal modal logic, **26**
 n th Cartesian power, **19**
 n -tuple, **19**
 nullary modality, **143**
 nullary modality semantics, **143**

O

one-to-oneness, **21**
 open world assumption, **102**
 operation, **23**
 associative, **23**
 commutative, **23**
 distributing, **23**
 idempotent, **23**
 n -ary, **23**
 operator, *see* operation
 optimization problem, **191**
 optional feature problem, **70**
 order, **22**
 overspecification, **70**

P

package, **32**
 pair, **19**
 parametric delta, **111**
 parametric delta model, **111**

parametric delta model evaluation, **111**
 parametric deltoid, **111**
 parametric product line evaluation, **112**
 parametric product line implementation, **112**
 parametric software delta, **112**
 parametric software delta evaluation, **112**
 partial function, **22**
 partial order, **22**
 partially defined delta, **44**
 partially functional deltoid, **59**
 patch, **53**
 pkgloader area, **127**
 pointcut, **5**
 positive natural numbers, **19**
 powerset, **18**
 predicate, **20**
 preimage, **20**
 premise, **21**
 preorder, **22**
 principal ideal, **151**
 product, **36**
 core, **103**
 initial, **57**
 product acceptance, **44**
 product family, *see* product line
 product formula, **136**
 product formula evaluation, **137**
 product line, **97**
 dynamic, **167**
 Editor, **7**
 Fredhopper Access Server, **161**
 Thesis, **120**
 product line correctness, **108**
 product line evaluation, **104**
 parametric, **112**
 product line implementation, **103**
 nested, **115**
 parametric, **112**
 product line implementation, dynamic, **175**
 product line specification, **108**
 structured, **210**
 product line unambiguity, **104**
 profile, **169**
 manual, **174**
 profile delta monoid, **171**

profile deltoid, **171**
 profile feature model, **174**
 program delta, **65**
 proper relation algebra, **49**
 propositional variable, **25**
 provability relation, **27**
 pure delta modeling, **152**
 pure delta oriented programming,
 152

Q

quadruple, **19**
 quantification and weaving, **60**
 quantity, **169**
 quark, **62**
 empty, **62**
 full, **62**
 global, **62**
 local, **62**
 simple, **62**
 quark deltoid, **63**
 quark model, **60**
 quintuple, **19**
 quotient algebra, **24**
 quotient deltoid, **38**
 quotient set, **23**

R

real numbers, **19**
 reconfiguration translation, **193**
 reflexivity, **21**
 relation, **20**
 antisymmetric, **21**
 asymmetric, **21**
 bijective, **21**
 discrete, **21**
 equivalence, **22**
 fully defined, **21**
 functional, **21**
 identity, **20**
 injective, **21**
 irreflexive, **21**
 n-ary, **20**
 one-to-one, **21**
 reflexive, **21**
 surjective, **21**
 symmetric, **21**
 total, **21**
 transitive, **21**
 uniquely defined, **21**

 well defined, **21**
 relation algebra, **24**
 proper, **49**
 relation algebra semantics, **49**
 relation composition, **20**
 relation diagram, **22**
 relation image, **20**
 relation inverse, **20**
 relation preimage, **20**
 relational deltoid, **58**
 rule-set, **172**
 rule-set implementation, **174**

S

satisfaction operator, **207**
 selected delta model, **103**
 self adaptive system, **194**
 semantic delta, **37**
 semantic delta equivalence, **48**
 semantic delta refinement, **47**
 semantics, **38**
 separation of concerns, **35**
 sequence, **19**
 sequence concatenation, **19**
 set, **18**
 set complement, **18**
 set difference, **18**
 set intersection, **18**
 set membership, **18**
 set union, **18**
 setting, **169**
 simple delta, **90**
 simple quark, **62**
 singleton transition function, **185**
 software class delta, **39**
 software delta, **39**
 empty, **54**
 invalid, **43**
 neutral, **54**
 software delta algebra, **56**
 software delta application, **43**
 software delta composition, **55**
 software delta consensus, **54**
 software delta equivalence, **44**
 software deltoid, **41**
 software deltoid evaluation, **41**
 software package delta, **39**
 software product line engineering,
 4
 sole derivation semantics, **75**

soundness, **27**
 SPLE, *see* software product line
 engineering
 stable configuration, **179**
 statement, **31**
 statement delta, **84**
 stone carving deltoid, **57**
 strategy, **180**
 strict order, **22**
 strong completeness, **27**
 structured product line specifica-
 tion, **210**
 stuck configuration, **28**
 subfeature, **150**
 subset, **18**
 superfeature, **150**
 superimposition, **60**
 superset, **18**
 surjectivity, **21**
 symmetric difference, **18**
 symmetric set difference, **18**
 symmetry, **21**
 syntactic delta equivalence, **53**
 syntactic delta refinement, **52**
 syntactic FgSD refinement, **86**
 syntactic software delta refinement,
 54
 syntax, **38**

T

target state, **180**
 test driven development, **107**
 T_EX, **119**
 Thesis product line, **120**
 three-way delta conflict, **110**
 token list, **119**
 totality, **21**
 trait, **66**
 transition, **28**
 environmental, **176**
 local, **176**
 transition function, **182**
 full difference, **183**
 minimal, **187**
 singleton, **185**
 transition relation, **28**
 DMW, **215**
 Mealy machine, **180**
 transitive closure, **21**
 transitivity, **21**

triple, **19**
 tuple, **19**
 type, **31**

U

unambiguity
 delta model, **79**
 unambiguous delta model, **79**
 undefined, **22**
 undefined delta, **44**
 uniform substitution, **26**
 unique derivation, **75**
 unique-definedness, **21**
 unstable configuration, **179**
 updated function, **22**

V

valuation function, **25, 107**
 variability, **5**
 viable conflict set, **214**
 viable feature combination, **213**
 viable feature combination order,
 213

W

well-definedness, **21**
 workflow state, **211**
 world, **25**

Bibliography of My Publications

- [1] D. Clarke, M. Helvensteijn, and I. Schaefer. “Abstract Delta Modeling”. In: 9th international conference on Generative Programming and Component Engineering. Vol. 46. SIGPLAN Notices. Eindhoven, The Netherlands: ACM, Oct. 10, 2010, pp. 13–22. ISBN: 9781-450-3015-4-1.
- [2] D. Clarke, M. Helvensteijn, and I. Schaefer. “Abstract Delta Modeling”. In: *Accepted to MSCS special issue* (2012).
- [3] F. d. Boer, M. Helvensteijn, and J. Winter. “A Modal Logic for Abstract Delta Modeling”. In: 16th International Software Product Line Conference. Vol. 2. SPLC ’12. Salvador, Brazil: ACM, Sept. 2, 2012, pp. 45–52. ISBN: 978-1-4503-1095-6.
- [4] M. Helvensteijn. “Abstract Delta Modeling: My Research Plan”. In: 16th International Software Product Line Conference. Vol. 2. SPLC ’12. Salvador, Brazil: ACM, Sept. 2, 2012, pp. 217–224. ISBN: 978-1-4503-1095-6.
- [5] M. Helvensteijn. “Delta Modeling Workflow”. In: 6th International Workshop on Variability Modelling of Software-intensive Systems. ACM International Conference Proceedings Series. Leipzig, Germany: ACM, 2012, pp. 129–137. ISBN: 978-1-4503-1058-1.
- [6] M. Helvensteijn. “Dynamic Delta Modeling”. In: 16th International Software Product Line Conference. Vol. 2. SPLC ’12. Salvador, Brazil: ACM, Sept. 2, 2012, pp. 127–134. ISBN: 978-1-4503-1095-6.
- [7] M. Helvensteijn, R. Muschevici, and P. Wong. “Delta Modeling in Practice, a Fredhopper Case Study”. In: 6th International Workshop on Variability Modelling of Software-intensive Systems. ACM International Conference Proceedings Series. Leipzig, Germany: ACM, 2012, pp. 139–148. ISBN: 978-1-4503-1058-1.
- [8] R. Hähnle et al. “HATS Abstract Behavioral Specification: The Architectural View”. In: *Formal Methods for Components and Objects*. Ed. by B. Beckert et al. Vol. 7542. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 109–132. ISBN: 978-3-642-35886-9, 978-3-642-35887-6.
- [9] D. Clarke, M. Helvensteijn, and I. Schaefer. *Abstract Delta Modeling (Technical Report)*. CW592. Dept. Computer Sciences, Katholieke Universiteit Leuven, Aug. 2010.

- [10] M. Helvensteijn. “The pkgloader and lt3graph Packages: Toward simple and powerful package management for LaTeX”. In: *TUGboat: The Communications of the TeX Users Group* Volume 35 (Issue 1 2014). Ed. by B. Beeton, pp. 47–51.

Main Bibliography

- [11] *Android*. In collab. with Google. 2008. URL: <http://www.android.com>.
- [12] S. Apel, D. Batory, and M. Rosenmüller. “On the Structure of Crosscutting Concerns: Using Aspects or Collaborations?” In: GPCE Workshop on Aspect-Oriented Product Line Engineering (AOPL). 2006.
- [13] S. Apel and C. Kästner. “An Overview of Feature-Oriented Software Development”. In: *Journal of Object Technology* 8 (5 July 2009), pp. 49–84. ISSN: 1660-1769.
- [14] S. Apel, C. Kästner, and D. Batory. “Program Refactoring Using Functional Aspects”. In: *Proceedings of the 7th international conference on Generative programming and component engineering*. GPCE ’08. New York, NY, USA: ACM, 2008, pp. 161–170. ISBN: 978-1-60558-267-2.
- [15] S. Apel, C. Kästner, and C. Lengauer. “FEATUREHOUSE: Language-Independent, Automated Software Composition”. In: IEEE 31st International Conference on Software Engineering. Vancouver, BC, May 16, 2009, pp. 221–231.
- [16] S. Apel, T. Leich, and G. Saake. “Aspectual Feature Modules”. In: *IEEE Transactions on Software Engineering* 34 (2 2008), pp. 162–180. ISSN: 0098-5589.
- [17] S. Apel et al. “An Algebraic Foundation for Automatic Feature-Based Program Synthesis”. In: *Science of Computer Programming* 75 (11 Nov. 1, 2010), pp. 1022–1047. ISSN: 0167-6423.
- [18] S. Apel et al. “Exploring Feature Interactions in the Wild”. In: (2013).
- [19] S. Apel et al. “Model Superimposition in Software Product Lines”. In: *Theory and Practice of Model Transformations*. 2nd International ICMT Conference. Ed. by R. F. Paige. Vol. 5563. Lecture Notes in Computer Science. Zurich, Switzerland: Springer Berlin Heidelberg, June 29, 2009, pp. 4–19. ISBN: 978-3-642-02407-8, 978-3-642-02408-5.
- [20] Apple. *iOS*. 2007. URL: <http://www.apple.com/ios>.
- [21] K. R. Apt, F. S. de Boer, and E.-R. Olderog. *Verification of Sequential and Concurrent Programs*. Dordrecht: Springer, 2009. 528 pp. ISBN: 9781848827448 184882744X.

- [22] C. Areces, P. Blackburn, and M. Marx. “Hybrid Logics: Characterization, Interpolation and Complexity”. In: *The Journal of Symbolic Logic* 66 (3 Sept. 2001), p. 977. ISSN: 00224812.
- [23] U. Aßmann. *Invasive Software Composition*. Springer, Feb. 27, 2003. 364 pp. ISBN: 9783540443858.
- [24] L. Baresi and C. Ghezzi. “The Disappearing Boundary Between Development-time and Run-time”. In: *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*. FoSER '10. New York, NY, USA: ACM, 2010, pp. 17–22. ISBN: 978-1-4503-0427-6.
- [25] D. Batory. “A Tutorial on Feature Oriented Programming and the AHEAD Tool Suite”. In: *Generative and Transformational Techniques in Software Engineering*. Lecture Notes in Computer Science (2006). Ed. by R. Lämmel, J. Saraiva, and J. Visser, pp. 3–35. ISSN: 0302-9743.
- [26] D. Batory. “Feature Models, Grammars, and Propositional Formulas”. In: *Proc. Int’l Software Product Line Conference (SPLC)*. 9th International Software Product Line Conference. Ed. by H. Obbink and K. Pohl. Vol. 3714. Lecture Notes in Computer Science. Rennes, France: Springer Berlin Heidelberg, 2005, pp. 7–20. ISBN: 978-3-540-28936-4, 978-3-540-32064-7.
- [27] D. Batory, D. Benavides, and A. Ruiz-Cortés. “Automated Analyses of Feature Models: Challenges Ahead”. In: *Communications of the ACM* 49 (12 Dec. 2006), pp. 45–47. ISSN: 0001-0782.
- [28] D. Batory and B. J. Geraci. “Composition Validation and Subjectivity in GenVoca Generators”. In: *IEEE Transactions on Software Engineering* 23 (2 1997), pp. 67–82. ISSN: 0098-5589.
- [29] D. Batory, P. Höfner, and J. Kim. “Feature Interactions, Products, and Composition”. In: 10th ACM international conference on Generative Programming and Component Engineering. Vol. 47. GPCE '11. ACM, 2011, pp. 13–22. ISBN: 978-1-4503-0689-8.
- [30] D. Batory and S. O’Malley. “The Design and Implementation of Hierarchical Software Systems with Reusable Components”. In: *ACM Transactions on Software Engineering Methodology* 1 (4 Oct. 1992), pp. 355–398. ISSN: 1049-331X.
- [31] D. Batory, J. N. Sarvela, and A. Rauschmayer. “Scaling Step-Wise Refinement”. In: *IEEE Transactions on Software Engineering* 30 (6 2004), pp. 355–371. ISSN: 0098-5589.
- [32] D. Batory and D. Smith. *Finite Map Spaces and Quarks: Algebras of Program Structure*. TR-07-66. Computer Science Department, University of Texas at Austin, 2007.
- [33] D. Batory et al. *Features, Modularity, and Variation Points*. TR-2147. University of Texas at Austin, June 6, 2013.
- [34] G. Bellin et al. “A Term Assignment for Dual Intuitionistic Logic”. In: LICS’05-IMLA’05 Workshop. 2005.
- [35] D. Benavides, S. Segura, and A. Ruiz-Cortés. “Automated Analysis of Feature Models 20 Years Later: a Literature Review”. In: *Information Systems* 35 (6 Sept. 2010), pp. 615–636. ISSN: 0306-4379.

- [36] Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions*. 1st. Texts in Theoretical Computer Science. Springer, June 24, 2004. 472 pp. ISBN: 978-3-662-07964-5, 978-3-540-20854-9, 978-3-642-05880-6.
- [37] L. Bettini, F. Damiani, and I. Schaefer. “Implementing Software Product Lines using Traits”. In: 2010 ACM Symposium on Applied Computing. ACM, 2010, pp. 2096–2102. ISBN: 978-1-60558-639-7.
- [38] G. Biegel and V. Cahill. “A Framework for Developing Mobile, Context-aware Applications”. In: *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. PerCom 2004*. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. PerCom 2004. 2004, pp. 361–365.
- [39] P. Blackburn. “Nominal Tense Logic”. In: *Notre Dame Journal of Formal Logic* 34 (1 1993), pp. 56–83.
- [40] P. Blackburn. “Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto”. In: *Logic Journal of IGPL* 8 (2000), pp. 339–365.
- [41] P. Blackburn, J. F. A. K. v. Benthem, and F. Wolter, eds. *Handbook of Modal Logic*. Vol. 3. Elsevier, Nov. 3, 2006. 1262 pp. ISBN: 9780080466668.
- [42] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001. 574 pp. ISBN: 9780521527149.
- [43] P. Blackburn and J. Seligman. “Hybrid Languages”. In: *Journal of Logic, Language and Information* 4 (3 Sept. 1, 1995), pp. 251–272. ISSN: 0925-8531, 1572-9583.
- [44] M. M. Bonsangue and J. N. Kok. “The Weakest Precondition Calculus: Recursion and Duality”. In: *Formal Aspects of Computing* 6 (1 Nov. 1, 1994), pp. 788–800. ISSN: 0934-5043, 1433-299X.
- [45] J. Bosch. *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. ACM Press/Addison-Wesley Publishing Co., 2000. ISBN: 0-201-67494-7.
- [46] Q. Boucher et al. “Introducing TVL, a Text-based Feature Modelling Language”. In: Fourth International Workshop on Variability Modelling of Software-intensive Systems. Linz, Austria, 2010, pp. 27–29.
- [47] G. Bracha and W. Cook. “Mixin-Based Inheritance”. In: *Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications. OOPSLA/E-COOP '90*. New York, NY, USA: ACM, 1990, pp. 303–311. ISBN: 0-89791-411-2.
- [48] G. Canfora et al. “In memory of Manny Lehman, ‘Father of Software Evolution’”. In: *Journal of Software Maintenance and Evolution: Research and Practice* 23 (3 2011), pp. 137–144. ISSN: 1532-0618.
- [49] D. Carlisle. *xii.tex*. Dec. 11, 1998. URL: <http://www.ctan.org/pkg/xii>.

- [50] B. H. C. Cheng et al. “Software Engineering for Self-Adaptive Systems: A Research Roadmap”. In: *Software Engineering for Self-Adaptive Systems*. Ed. by B. H. C. Cheng et al. Lecture Notes in Computer Science 5525. Springer Berlin Heidelberg, 2009, pp. 1–26. ISBN: 978-3-642-02160-2, 978-3-642-02161-9.
- [51] R. Chitchyan, J. Fabry, and L. Bergmans. “Aspects, Dependencies, and Interactions”. In: *Object-Oriented Technology. ECOOP 2006 Workshop Reader*. Ed. by M. Südholt and C. Consel. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 26–39. ISBN: 978-3-540-71772-0, 978-3-540-71774-4.
- [52] D. Clarke et al. “Variability Modelling in the ABS Language”. In: 9th international symposium on formal methods for components and objects (FMCO). Ed. by B. K. Aichernig, F. S. d. Boer, and M. M. Bonsangue. Vol. 6957. Lecture Notes in Computer Science. Graz, Austria: Springer Berlin Heidelberg, 2012, pp. 204–224. ISBN: 978-3-642-25270-9, 978-3-642-25271-6.
- [53] A. Classen, P. Heymans, and P.-Y. Schobbens. “What’s in a Feature: A Requirements Engineering Perspective”. In: *Fundamental Approaches to Software Engineering*. Ed. by J. L. Fiadeiro and P. Inverardi. Lecture Notes in Computer Science 4961. Springer Berlin Heidelberg, 2008, pp. 16–30. ISBN: 978-3-540-78742-6, 978-3-540-78743-3.
- [54] P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley Boston, 2002.
- [55] Colin Pickard. *product line*. In: July 25, 2011.
- [56] K. Czarnecki and M. Antkiewicz. “Mapping Features to Models: A Template Approach Based on Superimposed Variants”. In: 4th International Conference on Generative Programming and Component Engineering. Ed. by R. Glück and M. Lowry. Vol. 3676. Lecture Notes in Computer Science. Tallinn, Estonia: Springer Berlin Heidelberg, 2005, pp. 422–437. ISBN: 978-3-540-29138-1, 978-3-540-31977-1.
- [57] K. Czarnecki and U. W. Eisenecker. “Generative Programming”. In: Addison-Wesley, 2000, pp. 2–19.
- [58] K. Czarnecki, S. Helsen, and U. Eisenecker. “Staged Configuration Using Feature Models”. In: 3rd International Software Product Line Conference. Ed. by R. L. Nord. Vol. 3154. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 266–283. ISBN: 978-3-540-22918-6, 978-3-540-28630-1.
- [59] K. Czarnecki and C. H. P. Kim. “Cardinality-Based Feature Modeling and Constraints: A Progress Report”. In: International Workshop on Software Factories at OOPSLA’05. 2005, pp. 1–9.
- [60] K. Czarnecki and A. Wasowski. “Feature Diagrams and Logics: There and Back Again”. In: *Proceedings of the 11th International Software Product Line Conference*. 2007, pp. 23–34.

- [61] K. Czarnecki et al. “Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches”. In: *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*. 6th International Workshop on Variability Modeling of Software-Intensive Systems. ACM, 2012, pp. 173–182. ISBN: 978-1-4503-1058-1.
- [62] F. Damiani, C. Gladisch, and S. Tyszberowicz. “Refinement-based Testing of Delta-oriented Product Lines”. In: *The 2013 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools. PPPJ '13*. New York, NY, USA: ACM, 2013, pp. 135–140. ISBN: 978-1-4503-2111-2.
- [63] F. Damiani, L. Padovani, and I. Schaefer. “A Formal Foundation for Dynamic Delta-oriented Software Product Lines”. In: *Proceedings of the 11th International Conference on Generative Programming and Component Engineering. GPCE '12*. New York, NY, USA: ACM, 2012, pp. 1–10. ISBN: 978-1-4503-1129-8.
- [64] F. Damiani and I. Schaefer. “Dynamic Delta-Oriented Programming”. In: *15th International Software Product Line Conference*. Vol. 2. Munich, Germany: ACM, Aug. 22, 2011, p. 34. ISBN: 978-1-4503-0789-5.
- [65] S. Deelstra, M. Sinnema, and J. Bosch. “Product Derivation in Software Product Families: A Case Study”. In: *Journal of Systems and Software* 74 (2 2005), pp. 173–194. ISSN: 0164-1212.
- [66] A. van Deursen and P. Klint. “Domain-Specific Language Design Requires Feature Descriptions”. In: *Journal of Computing and Information Technology* 10 (1 2002), pp. 1–17.
- [67] S. Ducasse et al. “Traits: A Mechanism for Fine-Grained Reuse”. In: *ACM Trans. Program. Lang. Syst.* 28 (2 Mar. 2006), pp. 331–388. ISSN: 0164-0925.
- [68] Eclipse Foundation. *AspectJ*. 2001. URL: <http://eclipse.org/aspectj/>.
- [69] M. J. Fischer and R. E. Ladner. “Propositional Dynamic Logic of Regular Programs”. In: *Journal of Computer and System Sciences* 18 (2 Apr. 1979), pp. 194–211. ISSN: 0022-0000.
- [70] Florent Chervet. *The interfaces package for LaTeX*. 2010. URL: <http://www.ctan.org/pkg/interfaces>.
- [71] M. Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 1999. 474 pp. ISBN: 9780201485677.
- [72] G. Gargov and V. Goranko. “Modal Logic with Names”. In: *Journal of Philosophical Logic* 22 (6 Dec. 1, 1993), pp. 607–636. ISSN: 0022-3611, 1573-0433.
- [73] C. Ghezzi and A. M. Sharifloo. “Dealing with Non-Functional Requirements for Adaptive Systems via Dynamic Software Product-Lines”. In: *Software Engineering for Self-Adaptive Systems II*. Ed. by R. d. Lemos et al. Vol. 7475. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 191–213. ISBN: 978-3-642-35812-8, 978-3-642-35813-5.

- [74] d. G. Giuseppe. “Eliminating ”Converse” from Converse PDL”. In: *Journal of Logic, Language and Information* 5 (2 Apr. 1, 1996), pp. 193–208. ISSN: 0925-8531, 1572-9583.
- [75] M. L. Griss. “Implementing Product-Line Features by Composing Aspects”. In: the First Conference on Software Product Lines. Ed. by P. Donohoe. The Springer International Series in Engineering and Computer Science. Springer, 2000, pp. 271–288. ISBN: 978-1-4613-6949-3, 978-1-4615-4339-8.
- [76] A. Haber et al. “Engineering Delta Modeling Languages”. In: *Proceedings of the 17th International Software Product Line Conference*. SPLC ’13. ACM, 2013, pp. 22–31. ISBN: 978-1-4503-1968-3.
- [77] A. Haber et al. “Evolving Delta-Oriented Software Product Line Architectures”. In: *Large-Scale Complex IT Systems. Development, Operation and Management*. Ed. by R. Calinescu and D. Garlan. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 183–208. ISBN: 978-3-642-34058-1, 978-3-642-34059-8.
- [78] A. Haber et al. “First-Class Variability Modeling in Matlab/Simulink”. In: *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems*. VaMoS ’13. ACM, 2013. ISBN: 978-1-4503-1541-8.
- [79] A. Haber et al. “Towards a Family-based Analysis of Applicability Conditions in Architectural Delta Models”. In: VARY International Workshop affiliated with ACM/IEEE 14th International Conference on Model Driven Engineering Languages and Systems. 2011, pp. 43–52.
- [80] R. Hähnle. “HATS: Highly Adaptable and Trustworthy Software Using Formal Methods”. In: *Leveraging Applications of Formal Methods, Verification, and Validation*. Ed. by T. Margaria and B. Steffen. Vol. 6416. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 3–8. ISBN: 978-3-642-16560-3, 978-3-642-16561-0.
- [81] R. Hähnle and I. Schaefer. “A Liskov Principle for Delta-Oriented Programming”. In: *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change*. Ed. by T. Margaria and B. Steffen. Vol. 7609. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 32–46. ISBN: 978-3-642-34025-3, 978-3-642-34026-0.
- [82] S. Hallsteinsen et al. “Dynamic Software Product Lines”. In: *Computer* 41 (4 Apr. 2008), pp. 93–95. ISSN: 0018-9162.
- [83] P. R. Halmos. *Naive Set Theory*. Springer, 1960. 120 pp. ISBN: 9780387900926.
- [84] F. Heidenreich and C. Wende. “Bridging the Gap Between Features and Models”. In: Aspect-Oriented Product Line Engineering (AOPLE’07). 2007.
- [85] M. Helvensteijn. *Delta Profiles*. 2013. URL: <http://code.google.com/p/delta-profiles>.
- [86] M. Helvensteijn. *The lt3graph package for LaTeX3*. 2013. URL: <http://www.ctan.org/pkg/lt3graph>.

- [87] M. Helvensteijn. *The withargs package for LaTeX*. 2013. URL: <http://www.ctan.org/pkg/withargs>.
- [88] M. Hennessy. *The Semantics of Programming Languages: an Elementary Introduction using Structural Operational Semantics*. Wiley (Chichester England and New York), 1990.
- [89] R. E. Herrejon. “Understanding Feature Modularity”. University of Texas at Austin, 2006.
- [90] R. E. Herrejon, D. Batory, and W. Cook. “Evaluating Support for Features in Advanced Modularization Technologies”. In: ECOOP 2005 - Object-Oriented Programming. Ed. by A. P. Black. Vol. 3586. Lecture Notes in Computer Science. Springer, July 25, 2005, pp. 169–194. ISBN: 978-3-540-27992-1, 978-3-540-31725-8.
- [91] P. Heymans et al. “Evaluating Formal Properties of Feature Diagram Languages”. In: *IET Software* 2 (3 2008), pp. 281–302. ISSN: 1751-8806.
- [92] A. Heyting. *Intuitionism: An Introduction*. Vol. 41. Access Online via Elsevier, 1966.
- [93] C. Hoare. “A Calculus of Total Correctness for Communicating Processes”. In: *Science of Computer Programming* 1 (1–2 Oct. 1981), pp. 49–72. ISSN: 0167-6423.
- [94] C. A. R. Hoare. “An Axiomatic Basis for Computer Programming”. In: *Communications of the ACM* 12 (10 Oct. 1969), pp. 576–580. ISSN: 0001-0782.
- [95] D. R. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid*. New York: Basic Books, 1999. ISBN: 0394756827 9780394756820 0465026567 9780465026562.
- [96] W. L. Hürsch and C. V. Lopes. *Separation of Concerns*. College of Computer Science, Northeastern University, 1995.
- [97] J. Jacobson. *A Formalization of Darcs Patch Theory using Inverse Semigroups*. CAM report 09-83. UCLA, 2009.
- [98] N. Jacobson. *Lectures in Abstract Algebra*. Vol. 1. van Nostrand New York, 1951.
- [99] JetBrains. *IntelliJ IDEA*. Version 13. 2014. URL: <http://www.jetbrains.com/idea/>.
- [100] E. B. Johnsen et al. “ABS: A Core Language for Abstract Behavioral Specification”. In: *Formal Methods for Components and Objects*. 9th International Symposium on Formal Methods for Components and Objects. Ed. by B. K. Aichernig, F. S. de Boer, and M. M. Bonsangue. Vol. 6957. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 142–164. ISBN: 978-3-642-25270-9, 978-3-642-25271-6.
- [101] B. Jónnson and A. Tarski. “Boolean Algebras with Operators, Part II”. In: *American Journal of Mathematics* 74 (1 1952), pp. 127–162. ISSN: 00029327.
- [102] B. Jónsson and A. Tarski. “Representation Problems for Relation Algebras”. In: *Bull. Amer. Math. Soc* 54 (80 1948), p. 1192.

- [103] C. Julien and G.-C. Roman. “Egocentric Context-aware Programming in Ad Hoc Mobile Environments”. In: *Proceedings of the 10th ACM SIGSOFT symposium on Foundations of software engineering*. SIGSOFT ’02/FSE-10. New York, NY, USA: ACM, 2002, pp. 21–30. ISBN: 1-58113-514-9.
- [104] K. C. Kang, J. Lee, and P. Donohoe. “Feature-Oriented Product Line Engineering”. In: *IEEE Software* 5 (4 2002), pp. 58–65. ISSN: 0740-7459.
- [105] K. C. Kang et al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. SEI-90-TR-21. Carnegie Mellon University, 1990.
- [106] C. Kästner and S. Apel. “Type-Checking Software Product Lines - A Formal Approach”. In: 23rd IEEE/ACM International Conference on Automated Software Engineering. IEEE, 2008, pp. 258–267.
- [107] C. Kästner, S. Apel, and D. Batory. “A Case Study Implementing Features Using AspectJ”. In: 11th International Software Product Line Conference. IEEE, Sept. 10, 2007, pp. 223–232.
- [108] C. Kästner, S. Apel, and M. Kuhlemann. “Granularity in Software Product Lines”. In: 30th international conference on Software Engineering. ACM, 2008, pp. 311–320. ISBN: 978-1-60558-079-1.
- [109] C. Kästner, S. Apel, and K. Ostermann. “The Road to Feature Modularity?”. In: 15th International Software Product Line Conference. SPLC ’11. ACM, 2011, 5:1–5:8. ISBN: 978-1-4503-0789-5.
- [110] C. Kästner et al. “FeatureIDE: A Tool Framework for Feature-Oriented Software Development”. In: IEEE 31st International Conference on Software Engineering, 2009, pp. 611–614.
- [111] C. Kästner et al. “On the Impact of the Optional Feature Problem: Analysis and Case Studies”. In: The 13th International Software Product Line Conference. 2009, pp. 181–190.
- [112] G. Kiczales and M. Mezini. “Separation of Concerns with Procedures, Annotations, Advice and Pointcuts”. In: ECOOP 2005 - Object-Oriented Programming. Ed. by A. P. Black. Vol. 3586. Lecture Notes in Computer Science. Springer, 2005, pp. 195–213. ISBN: 978-3-540-27992-1, 978-3-540-31725-8.
- [113] G. Kiczales et al. “An Overview of AspectJ”. In: *ECOOP 2001 — Object-Oriented Programming*. ECOOP 2001 — Object-Oriented Programming. Ed. by J. L. Knudsen. Vol. 2072. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 327–354. ISBN: 978-3-540-42206-8, 978-3-540-45337-6.
- [114] G. Kiczales et al. “Aspect-Oriented Programming”. In: ECOOP’97 — Object-Oriented Programming. Ed. by M. Akşit and S. Matsuoka. Vol. 1241. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1997, pp. 220–242. ISBN: 978-3-540-63089-0, 978-3-540-69127-3.
- [115] D. E. Knuth. *The TeXbook*. In collab. with D. Bibby. Vol. 1993. Addison-Wesley Reading, MA, USA, 1986.
- [116] D. E. Knuth. *The Art of Computer Programming*. 3rd ed. Vol. Volume 1 (Fundamental Algorithms). Reading, Mass.: Addison-Wesley, July 7, 1997. 672 pp. ISBN: 978-0201896831.

- [117] C. W. Krueger. “New Methods in Software Product Line Development”. In: 10th International Software Product Line Conference. 2006, pp. 95–102.
- [118] L. Lamport. *LATEX: A Document Preparation System*. Pearson Education, Sept. 1, 1994. 292 pp. ISBN: 9788177584141.
- [119] M. M. Lehman. “Laws of Software Evolution Revisited”. In: *Software Process Technology*. Ed. by C. Montangero. Lecture Notes in Computer Science 1149. Springer, 1996, pp. 108–124. ISBN: 978-3-540-61771-6, 978-3-540-70676-2.
- [120] M. Lienhardt and D. Clarke. “Row Types for Delta-Oriented Programming”. In: 6th International Workshop on Variability Modeling of Software-Intensive Systems. ACM, 2012, pp. 121–128. ISBN: 978-1-4503-1058-1.
- [121] M. Lienhardt and D. Clarke. “Conflict Detection in Delta-Oriented Programming”. In: *Leveraging Applications of Formal Methods, Verification and Validation: Technologies for Mastering Change*. 5th international conference on Leveraging Applications of Formal Methods, Verification and Validation. Ed. by T. Margaria and B. Steffen. Vol. 7609. Lecture Notes in Computer Science. Heraklion, Crete, Greece: Springer Berlin Heidelberg, Oct. 15, 2012, pp. 178–192. ISBN: 978-3-642-34025-3, 978-3-642-34026-0.
- [122] F. v. d. Linden, K. Schmid, and E. Rommes. *Software Product Lines in Action: the Best Industrial Practice in Product Line Engineering*. Berlin: Springer, 2010. 333 pp. ISBN: 9783642090615 3642090613 9783540714378 3540714375.
- [123] S. Lipschutz and M. Lipson. *Schaum’s Outline of Discrete Mathematics*. Revised Third Edition. New York; London: Schaum, 2007. ISBN: 9780071615860 0071615865.
- [124] J. Liu, D. S. Batory, and S. Nedunuri. “Modeling Interactions in Feature Oriented Software Designs”. In: *FIW (2005)*, pp. 178–197.
- [125] J. Liu, D. Batory, and C. Lengauer. “Feature Oriented Refactoring of Legacy Applications”. In: 28th international conference on Software Engineering. ICSE ’06. ACM, 2006, pp. 112–121. ISBN: 1-59593-375-1.
- [126] N. Loughran and A. Rashid. “Framed Aspects: Supporting Variability and Configurability for AOP”. In: *Software Reuse: Methods, Techniques, and Tools*. 8th International ICSR Conference. Ed. by J. Bosch and C. Krueger. Vol. 3107. Lecture Notes in Computer Science. Madrid, Spain: Springer, July 5, 2004, pp. 127–140. ISBN: 978-3-540-22335-1, 978-3-540-27799-6.
- [127] R. C. Lyndon. “The Representation of Relational Algebras”. In: *Annals of Mathematics* 51 (3 May 1, 1950), pp. 707–729. ISSN: 0003-486X.
- [128] R. Maddux. “Some Sufficient Conditions for the Representability of Relation Algebras”. In: *Algebra Universalis* 8 (1 1978), pp. 162–172.
- [129] A. Marot and R. Wuyts. “Composing Aspects with Aspects”. In: 9th International Conference on Aspect-Oriented Software Development. AOSD ’10. ACM, 2010, pp. 157–168. ISBN: 978-1-60558-958-9.

- [130] R. C. Martin. *Agile Software Development: Principles, Patterns, and Practices*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2003. ISBN: 0135974445.
- [131] G. Mealy. “A Method for Synthesizing Sequential Circuits”. In: *Bell System Technical Journal* 34 (5 1955), pp. 1045–1079.
- [132] T. Mens, G. Taentzer, and O. Runge. “Detecting Structural Refactoring Conflicts Using Critical Pair Analysis”. In: *Electronic Notes in Theoretical Computer Science* 127 (3 Apr. 11, 2005), pp. 113–128. ISSN: 1571-0661.
- [133] M. Mezini and K. Ostermann. “Variability Management with Feature-Oriented Programming and Aspects”. In: *Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*. SIGSOFT ’04/FSE-12. ACM, 2004, pp. 127–136. ISBN: 1-58113-855-5.
- [134] Microsoft. *Windows Phone*. 2010. URL: <http://www.microsoft.com/windowsphone>.
- [135] A. Mikhajlova and E. Sekerinski. “Class Refinement and Interface Refinement in Object-Oriented Programs”. In: *FME’97: Industrial Applications and Strengthened Foundations of Formal Methods*. 4th International Symposium of Formal Methods Europe. Ed. by J. Fitzgerald, C. B. Jones, and P. Lucas. Vol. 1313. Lecture Notes in Computer Science. Graz, Austria: Springer Berlin Heidelberg, Sept. 15, 1997, pp. 82–101. ISBN: 978-3-540-63533-8, 978-3-540-69593-6.
- [136] R. Milner. *A Calculus of Communicating Systems*. Vol. 92. Lecture Notes in Computer Science. Springer Verlag, 1980. ISBN: 3-540-10235-3.
- [137] F. Mittelbach and C. Rowley. “The LATEX3 project”. In: *MAPS 93* (1993), pp. 95–99.
- [138] *Monkey patch*. In: *Wikipedia, the free encyclopedia*. In collab. with Wikipedia. 2014.
- [139] B. Moore et al. *Eclipse Development*. IBM Corporation, International Technical Support Organization, 2004.
- [140] S. Mosser et al. “Using Domain Features to Handle Feature Interactions”. In: *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*. VaMoS ’12. ACM, 2012, pp. 101–110. ISBN: 978-1-4503-1058-1.
- [141] R. Muschevici, D. Clarke, and J. Proença. “Executable Modelling of Dynamic Software Product Lines in the ABS Language”. In: *Proceedings of the 5th International Workshop on Feature-Oriented Software Development*. FOSD ’13. New York, NY, USA: ACM, 2013, pp. 17–24. ISBN: 978-1-4503-2168-6.
- [142] E. Y. Nakagawa, M. Becker, and J. C. Maldonado. “Towards a Process to Design Product Line Architectures Based on Reference Architectures”. In: *Proceedings of the 17th International Software Product Line Conference*. SPLC ’13. New York, NY, USA: ACM, 2013, pp. 157–161. ISBN: 978-1-4503-1968-3.

- [143] R. Nieuwenhuis, A. Oliveras, and C. Tinelli. “Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL (T)”. In: *Journal of the ACM* 53 (6 Nov. 2006), pp. 937–977. ISSN: 0004-5411.
- [144] N. Noda and T. Kishi. “Aspect-Oriented Modeling for Variability Management”. In: 12th International Software Product Line Conference. Sept. 8, 2008, pp. 213–222.
- [145] J. Oldevik, Ø. Haugen, and B. Møller-Pedersen. “Confluence in Domain-Independent Product Line Transformations”. In: 12th International Conference on Fundamental Approaches to Software Engineering: Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009. Ed. by M. Chechik and M. Wirsing. Vol. 5503. Lecture Notes in Computer Science. York, UK: Springer, Mar. 22, 2009, pp. 34–48. ISBN: 978-3-642-00592-3, 978-3-642-00593-0.
- [146] P. Oreizy et al. “An Architecture-based Approach to Self-adaptive Software”. In: *Intelligent Systems and Their Applications, IEEE* 14 (3 1999), pp. 54–62. ISSN: 1094-7167.
- [147] H. Ossher and P. Tarr. “Multi-Dimensional Separation of Concerns and the Hyperspace Approach”. In: *Software Architectures and Component Technology*. Ed. by M. Aksit. The Springer International Series in Engineering and Computer Science 648. Springer US, 2002, pp. 293–323. ISBN: 978-1-4613-5286-0, 978-1-4615-0883-0.
- [148] T. Panas, J. Andersson, and U. Aßmann. “The Editing Aspect of Aspects”. In: *Software Engineering and Applications*. ACTA Press, 2002.
- [149] S. Passy and T. Tinchev. “An Essay in Combinatory Dynamic Logic”. In: *Information and Computation* 93 (2 Aug. 1991), pp. 263–332. ISSN: 0890-5401.
- [150] G. Perrouin et al. “Reconciling Automation and Flexibility in Product Derivation”. In: 12th International Software Product Line Conference. 2008, pp. 339–348.
- [151] B. C. Pierce. *Advanced Topics In Types And Programming Languages*. MIT Press, 2005. 600 pp. ISBN: 9780262162289.
- [152] B. C. Pierce. *Types and Programming Languages*. MIT Press, 2002. 656 pp. ISBN: 9780262162098.
- [153] G. D. Plotkin. *A Structural Approach to Operational Semantics*. DAIMI FN-19. Aarhus University, 1981.
- [154] G. D. Plotkin. “The Origins of Structural Operational Semantics”. In: *The Journal of Logic and Algebraic Programming* 60–61 (July 2004), pp. 3–15. ISSN: 1567-8326.
- [155] K. Pohl, G. Böckle, and F. J. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, 2005.
- [156] C. Prehofer. “Feature-Oriented Programming: A Fresh Look at Objects”. In: *ECOOP’97 — Object-Oriented Programming*. Ed. by M. Aksit and S. Matsuoka. Vol. 1241. Lecture Notes in Computer Science. Springer, 1997, pp. 419–443. ISBN: 978-3-540-63089-0, 978-3-540-69127-3.

- [157] *product line*. In: red. by Douglas Harper.
- [158] Rob Kyff. “The Whole Ball Of Facts About Wax”. In: *Hartford Courant* (Apr. 17, 2002).
- [159] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language reference manual*. Addison-Wesley Longman Ltd., 1999. ISBN: 0-201-30998-X.
- [160] I. Schaefer. “Variability Modelling for Model-Driven Development of Software Product Lines”. In: 4th International Workshop on Variability Modelling of Software-intensive Systems (VaMoS 2010). 2010, pp. 85–92.
- [161] I. Schaefer, L. Bettini, and F. Damiani. “Compositional Type-Checking for Delta-oriented Programming”. In: 10th international conference on Aspect-oriented software development. New York, NY, USA: ACM, 2011, pp. 43–56. ISBN: 978-1-4503-0605-8.
- [162] I. Schaefer and F. Damiani. “Pure Delta-Oriented Programming”. In: 2nd International Workshop on Feature-Oriented Software Development. FOSD ’10. ACM, 2010, pp. 49–56. ISBN: 978-1-4503-0208-1.
- [163] I. Schaefer, A. Worret, and A. Poetzsch-Heffter. “A Model-Based Framework for Automated Product Derivation”. In: 1st International Workshop on Model-Driven Approaches in Software Product Line Engineering. Aug. 24, 2009, pp. 14–21.
- [164] I. Schaefer et al. “Delta-Oriented Programming of Software Product Lines”. In: *Software Product Lines: Going Beyond*. 14th International Software Product Line Conference. Ed. by J. Bosch and J. Lee. Vol. 6287. Lecture Notes in Computer Science. Jeju Island, South Korea: Springer, Sept. 13, 2010, pp. 77–91. ISBN: 978-3-642-15578-9, 978-3-642-15579-6.
- [165] N. Schärli et al. “Traits: Composable Units of Behaviour”. In: *ECOOP 2003 – Object-Oriented Programming*. Ed. by L. Cardelli. Vol. 2743. Lecture Notes in Computer Science. Springer, 2003, pp. 248–274. ISBN: 978-3-540-40531-3, 978-3-540-45070-2.
- [166] P.-Y. Schobbens, P. Heymans, and J.-C. Trigaux. “Feature Diagrams: A Survey and a Formal Semantics”. In: 14th IEEE International Requirements Engineering Conference. Los Alamitos, CA, USA: IEEE Computer Society, Sept. 11, 2006, pp. 139–148.
- [167] P.-Y. Schobbens et al. “Generic Semantics of Feature Diagrams”. In: *Computer Networks* 51 (2 Feb. 7, 2007), pp. 456–479. ISSN: 1389-1286.
- [168] R. Schöpf and R. Fairbairns. *CTAN - Comprehensive TeX Archive Network*. 1992. URL: <http://www.ctan.org>.
- [169] S. Schulze, O. Richers, and I. Schaefer. “Refactoring Delta-Oriented Software Product Lines”. In: *Proceedings of the 12th annual international conference on Aspect-oriented software development*. 2013, pp. 73–84.

- [170] L. Shen et al. “Towards Feature-Oriented Variability Reconfiguration in Dynamic Software Product Lines”. In: *Top Productivity through Software Reuse*. Ed. by K. Schmid. Vol. 6727. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 52–68. ISBN: 978-3-642-21346-5, 978-3-642-21347-2.
- [171] J. Sincero and W. Schröder-Preikschat. “The Linux Kernel Configurator as a Feature Modeling Tool”. In: Software Product Line Conference. 2008, pp. 257–260.
- [172] Y. Smaragdakis and D. Batory. “Mixin layers: an Object-Oriented Implementation Technique for Refinements and Collaboration-Based Designs”. In: *ACM Transactions on Software Engineering and Methodology*. Transactions of Software Engineering Methodology 11 (2 Apr. 2002), pp. 215–255. ISSN: 1049-331X.
- [173] Software Systems Generator Research Group. *AHEAD Tool Suite*. 2002. URL: <http://www.cs.utexas.edu/~schwartz/ATS/fopdocs/>.
- [174] Stan Lee. “Amazing Fantasy #15”. Aug. 10, 1962.
- [175] A. Tarski. “On the Calculus of Relations”. In: *The Journal of Symbolic Logic* 6 (3 Sept. 1941), pp. 73–89. ISSN: 0022-4812, 1943-5886.
- [176] The LaTeX Team. *The expl3 wrapper package for experimental LaTeX3*. 1990. URL: <http://www.ctan.org/pkg/expl3>.
- [177] V. A. Yankov (originator). *Brouwer Lattice*. In: *Encyclopedia of Mathematics*. 2011.
- [178] M. Völter and I. Groher. “Product Line Implementation using Aspect-Oriented and Model-Driven Software Development”. In: 11th International Software Product Line Conference. IEEE Computer Society, 2007, pp. 233–242. ISBN: 0-7695-2888-0.
- [179] S. Warner. *Modern algebra*. New York: Dover Publications, 1990. 818 pp. ISBN: 0486663418.
- [180] D. Weyns et al. “Claims and Supporting Evidence for Self-Adaptive Systems: A Literature Study”. In: 2012 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). 2012, pp. 89–98.
- [181] H. Zhang and S. Jarzabek. “An XVCL-based Approach to Software Product Line Development”. In: 15th International Conference on Software Engineering and Knowledge Engineering. 2003, pp. 267–275.

Titles in the IPA Dissertation Series since 2008

- W. Pieters.** *La Volonté Machinale: Understanding the Electronic Voting Controversy.* Faculty of Science, Mathematics and Computer Science, RU. 2008-01
- A.L. de Groot.** *Practical Automaton Proofs in PVS.* Faculty of Science, Mathematics and Computer Science, RU. 2008-02
- M. Bruntink.** *Renovation of Idiomatic Crosscutting Concerns in Embedded Systems.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2008-03
- A.M. Marin.** *An Integrated System to Manage Crosscutting Concerns in Source Code.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2008-04
- N.C.W.M. Braspenning.** *Model-based Integration and Testing of High-tech Multi-disciplinary Systems.* Faculty of Mechanical Engineering, TU/e. 2008-05
- M. Bravenboer.** *Exercises in Free Syntax: Syntax Definition, Parsing, and Assimilation of Language Conglomerates.* Faculty of Science, UU. 2008-06
- M. Torabi Dashti.** *Keeping Fairness Alive: Design and Formal Verification of Optimistic Fair Exchange Protocols.* Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2008-07
- I.S.M. de Jong.** *Integration and Test Strategies for Complex Manufacturing Machines.* Faculty of Mechanical Engineering, TU/e. 2008-08
- I. Hasuo.** *Tracing Anonymity with Coalgebras.* Faculty of Science, Mathematics and Computer Science, RU. 2008-09
- L.G.W.A. Cleophas.** *Tree Algorithms: Two Taxonomies and a Toolkit.* Faculty of Mathematics and Computer Science, TU/e. 2008-10
- I.S. Zapreev.** *Model Checking Markov Chains: Techniques and Tools.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-11
- M. Farshi.** *A Theoretical and Experimental Study of Geometric Networks.* Faculty of Mathematics and Computer Science, TU/e. 2008-12
- G. Gulesir.** *Evolvable Behavior Specifications Using Context-Sensitive Wildcards.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-13

- F.D. Garcia.** *Formal and Computational Cryptography: Protocols, Hashes and Commitments.* Faculty of Science, Mathematics and Computer Science, RU. 2008-14
- P. E. A. Dürr.** *Resource-based Verification for Robust Composition of Aspects.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-15
- E.M. Bortnik.** *Formal Methods in Support of SMC Design.* Faculty of Mechanical Engineering, TU/e. 2008-16
- R.H. Mak.** *Design and Performance Analysis of Data-Independent Stream Processing Systems.* Faculty of Mathematics and Computer Science, TU/e. 2008-17
- M. van der Horst.** *Scalable Block Processing Algorithms.* Faculty of Mathematics and Computer Science, TU/e. 2008-18
- C.M. Gray.** *Algorithms for Fat Objects: Decompositions and Applications.* Faculty of Mathematics and Computer Science, TU/e. 2008-19
- J.R. Calamé.** *Testing Reactive Systems with Data - Enumerative Methods and Constraint Solving.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-20
- E. Mumford.** *Drawing Graphs for Cartographic Applications.* Faculty of Mathematics and Computer Science, TU/e. 2008-21
- E.H. de Graaf.** *Mining Semi-structured Data, Theoretical and Experimental Aspects of Pattern Evaluation.* Faculty of Mathematics and Natural Sciences, UL. 2008-22
- R. Brijder.** *Models of Natural Computation: Gene Assembly and Membrane Systems.* Faculty of Mathematics and Natural Sciences, UL. 2008-23
- A. Koprowski.** *Termination of Rewriting and Its Certification.* Faculty of Mathematics and Computer Science, TU/e. 2008-24
- U. Khadim.** *Process Algebras for Hybrid Systems: Comparison and Development.* Faculty of Mathematics and Computer Science, TU/e. 2008-25
- J. Markovski.** *Real and Stochastic Time in Process Algebras for Performance Evaluation.* Faculty of Mathematics and Computer Science, TU/e. 2008-26
- H. Kastenbergh.** *Graph-Based Software Specification and Verification.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-27
- I.R. Buhan.** *Cryptographic Keys from Noisy Data Theory and Applications.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-28
- R.S. Marin-Perianu.** *Wireless Sensor Networks in Motion: Clustering Algorithms for Service Discovery and Provisioning.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2008-29
- M.H.G. Verhoef.** *Modeling and Validating Distributed Embedded Real-Time Control Systems.* Faculty of Science, Mathematics and Computer Science, RU. 2009-01
- M. de Mol.** *Reasoning about Functional Programs: Sparkle, a proof assistant for Clean.* Faculty of Science, Mathematics and Computer Science, RU. 2009-02
- M. Lormans.** *Managing Requirements Evolution.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-03

- M.P.W.J. van Osch.** *Automated Model-based Testing of Hybrid Systems.* Faculty of Mathematics and Computer Science, TU/e. 2009-04
- H. Sozer.** *Architecting Fault-Tolerant Software Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-05
- M.J. van Weerdenburg.** *Efficient Rewriting Techniques.* Faculty of Mathematics and Computer Science, TU/e. 2009-06
- H.H. Hansen.** *Coalgebraic Modelling: Applications in Automata Theory and Modal Logic.* Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-07
- A. Mesbah.** *Analysis and Testing of Ajax-based Single-page Web Applications.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-08
- A.L. Rodriguez Yakushev.** *Towards Getting Generic Programming Ready for Prime Time.* Faculty of Science, UU. 2009-9
- K.R. Olmos Joffré.** *Strategies for Context Sensitive Program Transformation.* Faculty of Science, UU. 2009-10
- J.A.G.M. van den Berg.** *Reasoning about Java programs in PVS using JML.* Faculty of Science, Mathematics and Computer Science, RU. 2009-11
- M.G. Khatib.** *MEMS-Based Storage Devices. Integration in Energy-Constrained Mobile Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-12
- S.G.M. Cornelissen.** *Evaluating Dynamic Analysis Techniques for Program Comprehension.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-13
- D. Bolzoni.** *Revisiting Anomaly-based Network Intrusion Detection Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-14
- H.L. Jonker.** *Security Matters: Privacy in Voting and Fairness in Digital Exchange.* Faculty of Mathematics and Computer Science, TU/e. 2009-15
- M.R. Czenko.** *TuLiP - Reshaping Trust Management.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-16
- T. Chen.** *Clocks, Dice and Processes.* Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-17
- C. Kaliszyk.** *Correctness and Availability: Building Computer Algebra on top of Proof Assistants and making Proof Assistants available over the Web.* Faculty of Science, Mathematics and Computer Science, RU. 2009-18
- R.S.S. O'Connor.** *Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory.* Faculty of Science, Mathematics and Computer Science, RU. 2009-19
- B. Ploeger.** *Improved Verification Methods for Concurrent Systems.* Faculty of Mathematics and Computer Science, TU/e. 2009-20
- T. Han.** *Diagnosis, Synthesis and Analysis of Probabilistic Models.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-21
- R. Li.** *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to*

Medical Image Analysis. Faculty of Mathematics and Natural Sciences, UL. 2009-22

J.H.P. Kwisthout. *The Computational Complexity of Probabilistic Networks*. Faculty of Science, UU. 2009-23

T.K. Cocx. *Algorithmic Tools for Data-Oriented Law Enforcement*. Faculty of Mathematics and Natural Sciences, UL. 2009-24

A.I. Baars. *Embedded Compilers*. Faculty of Science, UU. 2009-25

M.A.C. Dekker. *Flexible Access Control for Dynamic Collaborative Environments*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-26

J.F.J. Laros. *Metrics and Visualisation for Crime Analysis and Genomics*. Faculty of Mathematics and Natural Sciences, UL. 2009-27

C.J. Boogerd. *Focusing Automatic Code Inspections*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2010-01

M.R. Neuhäuser. *Model Checking Nondeterministic and Randomly Timed Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-02

J. Endrullis. *Termination and Productivity*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-03

T. Staijen. *Graph-Based Specification and Verification for Aspect-Oriented Languages*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-04

Y. Wang. *Epistemic Modelling and Protocol Dynamics*. Faculty of Science, UvA. 2010-05

J.K. Berendsen. *Abstraction, Prices and Probability in Model Checking Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2010-06

A. Nugroho. *The Effects of UML Modeling on the Quality of Software*. Faculty of Mathematics and Natural Sciences, UL. 2010-07

A. Silva. *Kleene Coalgebra*. Faculty of Science, Mathematics and Computer Science, RU. 2010-08

J.S. de Bruin. *Service-Oriented Discovery of Knowledge - Foundations, Implementations and Applications*. Faculty of Mathematics and Natural Sciences, UL. 2010-09

D. Costa. *Formal Models for Component Connectors*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-10

M.M. Jaghoori. *Time at Your Service: Schedulability Analysis of Real-Time and Distributed Services*. Faculty of Mathematics and Natural Sciences, UL. 2010-11

R. Bakhshi. *Gossiping Models: Formal Analysis of Epidemic Protocols*. Faculty of Sciences, Department of Computer Science, VUA. 2011-01

B.J. Arnoldus. *An Illumination of the Template Enigma: Software Code Generation with Templates*. Faculty of Mathematics and Computer Science, TU/e. 2011-02

E. Zambon. *Towards Optimal IT Availability Planning: Methods and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-03

L. Astefanoaei. *An Executable Theory of Multi-Agent Systems Refinement*. Faculty of Mathematics and Natural Sciences, UL. 2011-04

- J. Proença.** *Synchronous coordination of distributed components.* Faculty of Mathematics and Natural Sciences, UL. 2011-05
- A. Morali.** *IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-06
- M. van der Bijl.** *On changing models in Model-Based Testing.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-07
- C. Krause.** *Reconfigurable Component Connectors.* Faculty of Mathematics and Natural Sciences, UL. 2011-08
- M.E. Andrés.** *Quantitative Analysis of Information Leakage in Probabilistic and Nondeterministic Systems.* Faculty of Science, Mathematics and Computer Science, RU. 2011-09
- M. Atif.** *Formal Modeling and Verification of Distributed Failure Detectors.* Faculty of Mathematics and Computer Science, TU/e. 2011-10
- P.J.A. van Tilburg.** *From Computability to Executability – A process-theoretic view on automata theory.* Faculty of Mathematics and Computer Science, TU/e. 2011-11
- Z. Protic.** *Configuration management for models: Generic methods for model comparison and model co-evolution.* Faculty of Mathematics and Computer Science, TU/e. 2011-12
- S. Georgievska.** *Probability and Hiding in Concurrent Processes.* Faculty of Mathematics and Computer Science, TU/e. 2011-13
- S. Malakuti.** *Event Composition Model: Achieving Naturalness in Runtime Enforcement.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-14
- M. Raffelsieper.** *Cell Libraries and Verification.* Faculty of Mathematics and Computer Science, TU/e. 2011-15
- C.P. Tsirogiannis.** *Analysis of Flow and Visibility on Triangulated Terrains.* Faculty of Mathematics and Computer Science, TU/e. 2011-16
- Y.-J. Moon.** *Stochastic Models for Quality of Service of Component Connectors.* Faculty of Mathematics and Natural Sciences, UL. 2011-17
- R. Middelkoop.** *Capturing and Exploiting Abstract Views of States in OO Verification.* Faculty of Mathematics and Computer Science, TU/e. 2011-18
- M.F. van Amstel.** *Assessing and Improving the Quality of Model Transformations.* Faculty of Mathematics and Computer Science, TU/e. 2011-19
- A.N. Tamalet.** *Towards Correct Programs in Practice.* Faculty of Science, Mathematics and Computer Science, RU. 2011-20
- H.J.S. Basten.** *Ambiguity Detection for Programming Language Grammars.* Faculty of Science, UvA. 2011-21
- M. Izadi.** *Model Checking of Component Connectors.* Faculty of Mathematics and Natural Sciences, UL. 2011-22
- L.C.L. Kats.** *Building Blocks for Language Workbenches.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2011-23
- S. Kemper.** *Modelling and Analysis of Real-Time Coordination Patterns.* Faculty of Mathematics and Natural Sciences, UL. 2011-24

- J. Wang.** *Spiking Neural P Systems.* Faculty of Mathematics and Natural Sciences, UL. 2011-25
- A. Khosravi.** *Optimal Geometric Data Structures.* Faculty of Mathematics and Computer Science, TU/e. 2012-01
- A. Middelkoop.** *Inference of Program Properties with Attribute Grammars, Revisited.* Faculty of Science, UU. 2012-02
- Z. Hemel.** *Methods and Techniques for the Design and Implementation of Domain-Specific Languages.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-03
- T. Dimkov.** *Alignment of Organizational Security Policies: Theory and Practice.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-04
- S. Sedghi.** *Towards Provably Secure Efficiently Searchable Encryption.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-05
- F. Heidarian Dehkordi.** *Studies on Verification of Wireless Sensor Networks and Abstraction Learning for System Inference.* Faculty of Science, Mathematics and Computer Science, RU. 2012-06
- K. Verbeek.** *Algorithms for Cartographic Visualization.* Faculty of Mathematics and Computer Science, TU/e. 2012-07
- D.E. Nadales Agut.** *A Compositional Interchange Format for Hybrid Systems: Design and Implementation.* Faculty of Mechanical Engineering, TU/e. 2012-08
- H. Rahmani.** *Analysis of Protein-Protein Interaction Networks by Means of Annotated Graph Mining Algorithms.* Faculty of Mathematics and Natural Sciences, UL. 2012-09
- S.D. Vermolen.** *Software Language Evolution.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-10
- L.J.P. Engelen.** *From Napkin Sketches to Reliable Software.* Faculty of Mathematics and Computer Science, TU/e. 2012-11
- F.P.M. Stappers.** *Bridging Formal Models – An Engineering Perspective.* Faculty of Mathematics and Computer Science, TU/e. 2012-12
- W. Heijstek.** *Software Architecture Design in Global and Model-Centric Software Development.* Faculty of Mathematics and Natural Sciences, UL. 2012-13
- C. Kop.** *Higher Order Termination.* Faculty of Sciences, Department of Computer Science, VUA. 2012-14
- A. Osaiweran.** *Formal Development of Control Software in the Medical Systems Domain.* Faculty of Mathematics and Computer Science, TU/e. 2012-15
- W. Kuijper.** *Compositional Synthesis of Safety Controllers.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-16
- H. Beohar.** *Refinement of Communication and States in Models of Embedded Systems.* Faculty of Mathematics and Computer Science, TU/e. 2013-01
- G. Igna.** *Performance Analysis of Real-Time Task Systems using Timed Automata.* Faculty of Science, Mathematics and Computer Science, RU. 2013-02
- E. Zambon.** *Abstract Graph Transformation – Theory and Practice.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-03

- B. Lijnse.** *TOP to the Rescue – Task-Oriented Programming for Incident Response Applications.* Faculty of Science, Mathematics and Computer Science, RU. 2013-04
- G.T. de Koning Gans.** *Outsmarting Smart Cards.* Faculty of Science, Mathematics and Computer Science, RU. 2013-05
- M.S. Greiler.** *Test Suite Comprehension for Modular and Dynamic Systems.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-06
- L.E. Mamane.** *Interactive mathematical documents: creation and presentation.* Faculty of Science, Mathematics and Computer Science, RU. 2013-07
- M.M.H.P. van den Heuvel.** *Composition and synchronization of real-time components upon one processor.* Faculty of Mathematics and Computer Science, TU/e. 2013-08
- J. Businge.** *Co-evolution of the Eclipse Framework and its Third-party Plug-ins.* Faculty of Mathematics and Computer Science, TU/e. 2013-09
- S. van der Burg.** *A Reference Architecture for Distributed Software Deployment.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-10
- J.J.A. Keiren.** *Advanced Reduction Techniques for Model Checking.* Faculty of Mathematics and Computer Science, TU/e. 2013-11
- D.H.P. Gerrits.** *Pushing and Pulling: Computing push plans for disk-shaped robots, and dynamic labelings for moving points.* Faculty of Mathematics and Computer Science, TU/e. 2013-12
- M. Timmer.** *Efficient Modelling, Generation and Analysis of Markov Automata.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-13
- M.J.M. Roeloffzen.** *Kinetic Data Structures in the Black-Box Model.* Faculty of Mathematics and Computer Science, TU/e. 2013-14
- L. Lensink.** *Applying Formal Methods in Software Development.* Faculty of Science, Mathematics and Computer Science, RU. 2013-15
- C. Tankink.** *Documentation and Formal Mathematics — Web Technology meets Proof Assistants.* Faculty of Science, Mathematics and Computer Science, RU. 2013-16
- C. de Gouw.** *Combining Monitoring with Run-time Assertion Checking.* Faculty of Mathematics and Natural Sciences, UL. 2013-17
- J. van den Bos.** *Gathering Evidence: Model-Driven Software Engineering in Automated Digital Forensics.* Faculty of Science, UvA. 2014-01
- D. Hadziosmanovic.** *The Process Matters: Cyber Security in Industrial Control Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-02
- A.J.P. Jeckmans.** *Cryptographically-Enhanced Privacy for Recommender Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-03
- C.-P. Bezemer.** *Performance Optimization of Multi-Tenant Software Systems.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2014-04
- T.M. Ngo.** *Qualitative and Quantitative Information Flow Analysis for Multi-threaded Programs.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-05

A.W. Laarman. *Scalable Multi-Core Model Checking.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-06

J. Winter. *Coalgebraic Characterizations of Automata-Theoretic Classes.* Faculty of Science, Mathematics and Computer Science, RU. 2014-07

W. Meulemans. *Similarity Measures and Algorithms for Cartographic Schematization.* Faculty of Mathematics and Computer Science, TU/e. 2014-08

A.F.E. Belinfante. *JTorX: Exploring Model-Based Testing.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-09

A.P. van der Meer. *Domain Specific Languages and their Type Systems.* Faculty of Mathematics and Computer Science, TU/e. 2014-10

B.N. Vasilescu. *Social Aspects of Collaboration in Online Software Communities.* Faculty of Mathematics and Computer Science, TU/e. 2014-11

F.D. Aarts. *Tomte: Bridging the Gap between Active Learning and Real-World Systems.* Faculty of Science, Mathematics and Computer Science, RU. 2014-12

N. Noroozi. *Improving Input-Output Conformance Testing Theories.* Faculty of Mathematics and Computer Science, TU/e. 2014-13

M. Helvensteijn. *Abstract Delta Modeling: Software Product Lines and Beyond.* Faculty of Mathematics and Natural Sciences, UL. 2014-14