Bibliography

[Albert et al., 2015a] Albert, E., Arenas, P., Correas, J., Genaim, S., Gómez-Zamalloa, M., Martin-Martin, E., Puebla, G., and Román-Díez, G. (2015a). Resource Analysis: From Sequential to Concurrent and Distributed Programs. In *FM 2015: Formal Methods*, volume 9109, pages 3–17. Springer International Publishing, Cham.

[Albert et al., 2015b] Albert, E., Arenas, P., Correas, J., Genaim, S., Gómez-Zamalloa, M., Puebla, G., and Román-Díez, G. (2015b). Object-sensitive cost analysis for concurrent objects. *Software Testing, Verification and Reliability*, 25(3):218–271.

[Albert et al., 2014a] Albert, E., Arenas, P., Flores-Montoya, A., Genaim, S., Gómez-Zamalloa, M., Martin-Martin, E., Puebla, G., and Román-Díez, G. (2014a). SACO: Static Analyzer for Concurrent Objects. In *Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science, pages 562–567. Springer, Berlin, Heidelberg.

[Albert et al., 2012] Albert, E., Arenas, P., and Gómez-Zamalloa, M. (2012). Symbolic Execution of Concurrent Objects in CLP. In *Practical Aspects of Declarative Languages*, volume 7149, pages 123–137. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Albert et al., 2015c] Albert, E., Arenas, P., and Gómez-Zamalloa, M. (2015c). Test Case Generation of Actor Systems. In *Automated Technology for Verification and Analysis*, Lecture Notes in Computer Science, pages 259–275. Springer, Cham.

[Albert et al., 2016] Albert, E., Bezirgiannis, N., Boer, F. d., and Martin-Martin, E. (2016). A Formal, Resource Consumption-Preserving Translation of Actors to Haskell. In *Logic-Based Program Synthesis and Transformation*, Lecture Notes in Computer Science, pages 21–37. Springer, Cham.

[Albert et al., 2014b] Albert, E., Boer, F. S. d., Hähnle, R., Johnsen, E. B., Schlatte, R., Tarifa, S. L. T., and Wong, P. Y. H. (2014b). Formal modeling and analysis of resource management for cloud architectures: an industrial case study using Real-Time ABS. *Service Oriented Computing and Applications*, 8(4):323–339.

[Albert et al., 2013] Albert, E., Flores-Montoya, A., Genaim, S., and Martin-Martin, E. (2013). Termination and Cost Analysis of Loops with Concurrent Interleavings. In *Automated Technology for Verification and Analysis*, volume 8172, pages 349–364. Springer International Publishing, Cham.

[Azadbakht et al., 2017a] Azadbakht, K., Bezirgiannis, N., and Boer, F. S. d. (2017a). Distributed Network Generation Based on Preferential Attachment in ABS. In *SOFSEM 2017: Theory and Practice of Computer Science*, Lecture Notes in Computer Science, pages 103–115. Springer, Cham.

[Azadbakht et al., 2017b] Azadbakht, K., Bezirgiannis, N., and Boer, F. S. d. (2017b). On Futures for Streaming Data in ABS. In *Formal Techniques for Distributed Objects, Components, and Systems*, Lecture Notes in Computer Science, pages 67–73. Springer, Cham.

[Azadbakht et al., 2016] Azadbakht, K., Bezirgiannis, N., Boer, F. S. d., and Aliakbary, S. (2016). A High-level and Scalable Approach for Generating Scale-free Graphs Using Active Objects. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, pages 1244–1250, New York, NY, USA. ACM.

[Barabási and Albert, 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512.

[Bellifemine et al., 1999] Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE–A FIPA-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London.

[Berry and Boudol, 1990] Berry, G. and Boudol, G. (1990). The Chemical Abstract Machine. In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '90, pages 81–94, New York, NY, USA. ACM.

[Bezirgiannis and Boer, 2016] Bezirgiannis, N. and Boer, F. d. (2016). ABS: A High-Level Modeling Language for Cloud-Aware Programming. In *SOFSEM 2016: Theory and Practice of Computer Science*, Lecture Notes in Computer Science, pages 433–444. Springer, Berlin, Heidelberg.

[Bezirgiannis et al., 2017] Bezirgiannis, N., Boer, F. d., and Gouw, S. d. (2017). Human-in-the-Loop Simulation of Cloud Services. In *Service-Oriented and Cloud Computing*, Lecture Notes in Computer Science, pages 143–158. Springer, Cham.

[Bijo et al., 2016] Bijo, S., Johnsen, E. B., Pun, K. I., and Tarifa, S. L. T. (2016). A Maude Framework for Cache Coherent Multicore Architectures. In *Rewriting Logic and Its Applications*, Lecture Notes in Computer Science, pages 47–63. Springer, Cham.

[Bjørk et al., 2013] Bjørk, J., Boer, F. S. d., Johnsen, E. B., Schlatte, R., and Tarifa, S. L. T. (2013). User-defined schedulers for real-time concurrent objects. *Innovations in Systems and Software Engineering*, 9(1):29–43.

[Black et al., 2002] Black, A. P., Carlsson, M., Jones, M. P., Kieburtz, R., and Nordlander, J. (2002). Timber: A programming language for real-time embedded systems. Technical report.

[Boer et al., 2007] Boer, F. S. d., Clarke, D., and Johnsen, E. B. (2007). A complete guide to the future. In *Programming Languages and Systems*, pages 316–330. Springer.

[Boer and Gouw, 2014] Boer, F. S. d. and Gouw, S. d. (2014). Combining Monitoring with Run-Time Assertion Checking. In *Formal Methods for Executable Software Models*, Lecture Notes in Computer Science, pages 217–262. Springer, Cham.

[Boer et al., 2013] Boer, F. S. d., Gouw, S. d., and Wong, P. Y. H. (2013). Run-Time Verification of Coboxes. In *Software Engineering and Formal Methods*, Lecture Notes in Computer Science, pages 259–273. Springer, Berlin, Heidelberg.

[Brandauer et al., 2015] Brandauer, S., Castegren, E., Clarke, D., Fernandez-Reyes, K., Johnsen, E. B., Pun, K. I., Tarifa, S. L. T., Wrigstad, T., and Yang, A. M. (2015). Parallel Objects for Multicores: A Glimpse at the Parallel Language Encore. In *Formal Methods for Multicore Programming*, Lecture Notes in Computer Science, pages 1–56. Springer, Cham.

[Calheiros et al., 2011] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., and Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50.

[Claessen and Hughes, 2011] Claessen, K. and Hughes, J. (2011). QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. *SIGPLAN Not.*, 46(4):53–64.

[Clarke et al., 2010] Clarke, D., Helvensteijn, M., and Schaefer, I. (2010). Abstract Delta Modeling. In *Proceedings of the Ninth International Conference on Generative Programming and Component Engineering*, GPCE '10, pages 13–22, New York, NY, USA. ACM.

[Clebsch et al., 2015] Clebsch, S., Drossopoulou, S., Blessing, S., and McNeil, A. (2015). Deny Capabilities for Safe, Fast Actors. In *Proceedings of the 5th International Workshop on Programming Based on Actors, Agents, and Decentralized Control*, AGERE! 2015, pages 1–12, New York, NY, USA. ACM.

[Dean and Ghemawat, 2008] Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107.

[Din et al., 2015] Din, C. C., Bubel, R., and Hähnle, R. (2015). KeY-ABS: A Deductive Verification Tool for the Concurrent Modelling Language ABS. In *Automated Deduction - CADE-25*, Lecture Notes in Computer Science, pages 517–526. Springer, Cham.

[Din et al., 2017] Din, C. C., Owe, O., and Bubel, R. (2017). Runtime Assertion Checking and Theorem Proving for Concurrent and Distributed Systems. pages 480–487.

[Doménech et al., 2017] Doménech, J., Genaim, S., Johnsen, E. B., and Schlatte, R. (2017). EasyInterface: A Toolkit for Rapid Development of GUIs for Research Prototype Tools. In *Fundamental Approaches to Software Engineering*, Lecture Notes in Computer Science, pages 379–383. Springer, Berlin, Heidelberg.

[Eisenberg, 2015] Eisenberg, R. (2015). System FC, as implemented in GHC.

[Epstein et al., 2011] Epstein, J., Black, A. P., and Peyton-Jones, S. (2011). Towards Haskell in the cloud. In *ACM SIGPLAN Notices*, volume 46, pages 118–129. ACM.

[Flanagan and Felleisen, 1995] Flanagan, C. and Felleisen, M. (1995). The semantics of future and its use in program optimization. pages 209–220. ACM Press.

[Giachino et al., 2016a] Giachino, E., Gouw, S. d., Laneve, C., and Nobakht, B. (2016a). Statically and Dynamically Verifiable SLA Metrics. In *Theory and Practice of Formal Methods*, Lecture Notes in Computer Science, pages 211–225. Springer, Cham.

[Giachino et al., 2014] Giachino, E., Kobayashi, N., and Laneve, C. (2014). Deadlock Analysis of Unbounded Process Networks. In *CONCUR 2014 – Concurrency Theory*, volume 8704, pages 63–77. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Giachino et al., 2016b] Giachino, E., Laneve, C., and Lienhardt, M. (2016b). A framework for deadlock detection in core ABS. *Software & Systems Modeling*, 15(4):1013–1048.

[Gibbons, 2007] Gibbons, J. (2007). Datatype-Generic Programming. In *Datatype-Generic Programming*, Lecture Notes in Computer Science, pages 1–71. Springer, Berlin, Heidelberg.

[Göri et al., 2014] Göri, G., Johnsen, E. B., Schlatte, R., and Stolz, V. (2014). Erlang-Style Error Recovery for Concurrent Objects with Cooperative Scheduling. In *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications*, Lecture Notes in Computer Science, pages 5–21. Springer, Berlin, Heidelberg.

[Gouw et al., 2016] Gouw, S. d., Mauro, J., Nobakht, B., and Zavattaro, G. (2016). Declarative Elasticity in ABS. In *Service-Oriented and Cloud Computing*, Lecture Notes in Computer Science, pages 118–134. Springer, Cham.

[Hewitt et al., 1973] Hewitt, C., Bishop, P., and Steiger, R. (1973). A Universal Modular ACTOR Formalism for Artificial Intelligence. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, IJCAI'73, pages 235–245, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Holzmann, 2003] Holzmann, G. (2003). *Spin Model Checker, the: Primer and Reference Manual*. Addison-Wesley Professional, first edition.

[Imam and Sarkar, 2014] Imam, S. M. and Sarkar, V. (2014). Savina - An Actor Benchmark Suite: Enabling Empirical Evaluation of Actor Libraries. In *Proceedings of the 4th International Workshop on Programming Based on Actors Agents & Decentralized Control*, AGERE! '14, pages 67–80, New York, NY, USA. ACM.

[Jefferson and Sowizral, 1985] Jefferson, D. and Sowizral, H. (1985). Fast concurrent simulation using the Time Warp mechanism. In *SCS Conf. Distributed Simulation*, pages 63–69.

[Johnsen et al., 2010a] Johnsen, E. B., Hähnle, R., Schäfer, J., Schlatte, R., and Steffen, M. (2010a). ABS: A Core Language for Abstract Behavioral Specification. In *Formal Methods for Components and Objects*, Lecture Notes in Computer Science, pages 142–164. Springer, Berlin, Heidelberg.

[Johnsen et al., 2010b] Johnsen, E. B., Owe, O., Schlatte, R., and Tarifa, S. L. T. (2010b). Validating Timed Models of Deployment Components with Parametric Concurrency. In *Formal Verification of Object-Oriented Software*, Lecture Notes in Computer Science, pages 46–60. Springer, Berlin, Heidelberg.

[Johnsen et al., 2006] Johnsen, E. B., Owe, O., and Yu, I. C. (2006). Creol: A type-safe object-oriented model for distributed concurrent systems. *Theoretical Computer Science*, 365(1):23–66.

[Johnsen et al., 2012] Johnsen, E. B., Schlatte, R., and Tarifa, S. L. T. (2012). Modeling Resource-Aware Virtualized Applications for the Cloud in Real-Time ABS. In *Formal Methods and Software Engineering*, Lecture Notes in Computer Science, pages 71–86. Springer, Berlin, Heidelberg.

[Kiselyov and Laemmel, 2005] Kiselyov, O. and Laemmel, R. (2005). Haskell's overlooked object system. *arXiv:cs/0509027*. arXiv: cs/0509027.

[Kiselyov et al., 2004] Kiselyov, O., Lämmel, R., and Schupke, K. (2004). Strongly typed heterogeneous collections. In *Proceedings of the 2004 ACM SIGPLAN workshop on Haskell*, pages 96–107. ACM.

[Kliazovich et al., 2010] Kliazovich, D., Bouvry, P., Audzevich, Y., and Khan, S. U. (2010). GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers. pages 1–5. IEEE.

[Knuth, 1973] Knuth, D. E. (1973). *The art of computer programming*. Addison-Wesley series in computer science and information processing. Addison-Wesley Pub. Co, Reading, Mass.

[Lanese et al., 2014] Lanese, I., Lienhardt, M., Bravetti, M., Johnsen, E. B., Schlatte, R., Stolz, V., and Zavattaro, G. (2014). Fault Model Design Space for Cooperative Concurrency. In *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications*, Lecture Notes in Computer Science, pages 22–36. Springer, Berlin, Heidelberg.

[Long et al., 2005] Long, Q., Liu, Z., Li, X., and Jifeng, H. (2005). Consistent code generation from UML models. In *2005 Australian Software Engineering Conference*, pages 23–30.

[Magalhães et al., 2010] Magalhães, J. P., Dijkstra, A., Jeuring, J., and Löh, A. (2010). A Generic Deriving Mechanism for Haskell. In *Proceedings of the Third ACM Haskell Symposium on Haskell*, Haskell '10, pages 37–48, New York, NY, USA. ACM.

[McBride, 2000] McBride, C. (2000). Dependently Typed Functional Programs and their Proofs.

[Misra, 1986] Misra, J. (1986). Distributed discrete-event simulation. *ACM Computing Surveys (CSUR)*, 18(1):39–65.

[Moreira et al., 2010] Moreira, T. G., Wehrmeister, M. A., Pereira, C. E., Pétin, J. F., and Levrat, E. (2010). Automatic code generation for embedded systems: From UML specifications to VHDL code. In *2010 8th IEEE International Conference on Industrial Informatics*, pages 1085–1090.

[Nakata and Saar, 2013] Nakata, K. and Saar, A. (2013). Compiling Cooperative Task Management to Continuations. In *Fundamentals of Software Engineering*, pages 95–110. Springer.

[Nipkow et al., 2002] Nipkow, T., Paulson, L. C., and Wenzel, M. (2002). *Isabelle/HOL: a proof assistant for higher-order logic*. Number 2283 in Lecture notes in computer science. Springer, Berlin ; New York.

[Noll, 2001] Noll, T. (2001). A Rewriting Logic Implementation of Erlang. *Electronic Notes in Theoretical Computer Science*, 44(2):206–224.

[Nordlander, 2002] Nordlander, J. (2002). Polymorphic subtyping in O'Haskell. *Science of Computer Programming*, 43(2–3):93–127.

[Núñez et al., 2012] Núñez, A., Vázquez-Poletti, J. L., Caminero, A. C., Castañé, G. G., Carretero, J., and Llorente, I. M. (2012). iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator. *Journal of Grid Computing*, 10(1):185–209.

[Palacios et al., 2015] Palacios, A., Vidal, G., and Herbstritt, M. (2015). Towards Modelling Actor-Based Concurrency in Term Rewriting. Technical report, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany.

[Peyton Jones, 2003] Peyton Jones, S. L., editor (2003). *Haskell 98 language and libraries: the revised report*. Cambridge University Press, Cambridge, U.K. ; New York. OCLC: ocm51271691.

[Schäfer and Poetzsch-Heffter, 2010] Schäfer, J. and Poetzsch-Heffter, A. (2010). JCoBox: Generalizing Active Objects to Concurrent Components. In *ECOOP 2010 – Object-Oriented Programming*, Lecture Notes in Computer Science, pages 275–299. Springer, Berlin, Heidelberg.

[Sirjani et al., 2004] Sirjani, M., Movaghar, A., Shali, A., Boer, D., and S, F. (2004). Modeling and Verification of Reactive Systems using Rebeca. *Fundamenta Informaticae*, 63(4):385–410.

[Srinivasan and Mycroft, 2008] Srinivasan, S. and Mycroft, A. (2008). Kilim: Isolation-Typed Actors for Java. In *ECOOP 2008 – Object-Oriented Programming*, Lecture Notes in Computer Science, pages 104–128. Springer, Berlin, Heidelberg.

[Sulzmann et al., 2007] Sulzmann, M., Chakravarty, M. M., Jones, S. P., and Donnelly, K. (2007). System F with type equality coercions. In *Proceedings of the 2007 ACM SIGPLAN international workshop on Types in languages design and implementation*, pages 53–66. ACM.

[Tarau, 2011] Tarau, P. (2011). Coordination and Concurrency in Multi-engine Prolog. In *Coordination Models and Languages*, Lecture Notes in Computer Science, pages 157–171. Springer, Berlin, Heidelberg.

[Vidal, 2014] Vidal, G. (2014). Towards Erlang Verification by Term Rewriting. In Gupta, G. and Peña, R., editors, *Logic-Based Program Synthesis and Transformation*, volume 8901, pages 109–126. Springer International Publishing, Cham.

[Walker and Runciman, 2015] Walker, M. and Runciman, C. (2015). Déjà fu: a concurrency testing library for haskell. pages 141–152. ACM Press.

[Wong et al., 2012] Wong, P. Y., Albert, E., Muschevici, R., Proença, J., Schäfer, J., and Schlatte, R. (2012). The ABS tool suite: modelling, executing and analysing distributed adaptable object-oriented systems. *International Journal on Software Tools for Technology Transfer*, 14(5):567–588.

[Wong et al., 2015] Wong, P. Y. H., Bubel, R., Boer, F. S. d., Gómez-Zamalloa, M., Gouw, S. d., Hähnle, R., Meinke, K., and Sindhu, M. A. (2015). Testing abstract behavioral specifications. *International Journal on Software Tools for Technology Transfer*, 17(1):107–119.

Titles in the IPA Dissertation Series since 2015

G. Alpár. *Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World.* Faculty of Science, Mathematics and Computer Science, RU. 2015-01

A.J. van der Ploeg. *Efficient Abstractions for Visualization and Interaction.* Faculty of Science, UvA. 2015-02

R.J.M. Theunissen. *Supervisory Control in Health Care Systems.* Faculty of Mechanical Engineering, TU/e. 2015-03

T.V. Bui. *A Software Architecture for Body Area Sensor Networks: Flexibility and Trustworthiness.* Faculty of Mathematics and Computer Science, TU/e. 2015-04

A. Guzzi. *Supporting Developers' Teamwork from within the IDE.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-05

T. Espinha. *Web Service Growing Pains: Understanding Services and Their Clients.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-06

S. Dietzel. *Resilient In-network Aggregation for Vehicular Networks.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2015-07

E. Costante. *Privacy throughout the Data Cycle.* Faculty of Mathematics and Computer Science, TU/e. 2015-08

S. Cranen. *Getting the point — Obtaining and understanding fixpoints in model checking.* Faculty of Mathematics and Computer Science, TU/e. 2015-09

R. Verdult. *The (in)security of proprietary cryptography.* Faculty of Science, Mathematics and Computer Science, RU. 2015-10

J.E.J. de Ruiter. *Lessons learned in the analysis of the EMV and TLS security protocols.* Faculty of Science, Mathematics and Computer Science, RU. 2015-11

Y. Dajsuren. *On the Design of an Architecture Framework and Quality Evaluation for Automotive Software Systems.* Faculty of Mathematics and Computer Science, TU/e. 2015-12

J. Bransen. *On the Incremental Evaluation of Higher-Order Attribute Grammars.* Faculty of Science, UU. 2015-13

S. Picek. *Applications of Evolutionary Computation to Cryptology.* Faculty of Science, Mathematics and Computer Science, RU. 2015-14

C. Chen. *Automated Fault Localization for Service-Oriented Software Systems.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-15

S. te Brinke. *Developing Energy-Aware Software.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2015-16

R.W.J. Kersten. *Software Analysis Methods for Resource-Sensitive Systems.* Faculty of Science, Mathematics and Computer Science, RU. 2015-17

J.C. Rot. *Enhanced coinduction.* Faculty of Mathematics and Natural Sciences, UL. 2015-18

M. Stolikj. *Building Blocks for the Internet of Things.* Faculty of Mathematics and Computer Science, TU/e. 2015-19

D. Gebler. *Robust SOS Specifications of Probabilistic Processes.* Faculty of Sciences, Department of Computer Science, VUA. 2015-20

M. Zaharieva-Stojanovski. *Closer to Reliable Software: Verifying functional behaviour of concurrent programs.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2015-21

R.J. Krebbers. *The C standard formalized in Coq.* Faculty of Science, Mathematics and Computer Science, RU. 2015-22

R. van Vliet. *DNA Expressions – A Formal Notation for DNA.* Faculty of Mathematics and Natural Sciences, UL. 2015-23

S.-S.T.Q. Jongmans. *Automata-Theoretic Protocol Programming.* Faculty of Mathematics and Natural Sciences, UL. 2016-01

S.J.C. Joosten. *Verification of Interconnects.* Faculty of Mathematics and Computer Science, TU/e. 2016-02

M.W. Gazda. *Fixpoint Logic, Games, and Relations of Consequence.* Faculty of Mathematics and Computer Science, TU/e. 2016-03

S. Keshishzadeh. *Formal Analysis and Verification of Embedded Systems for Healthcare.* Faculty of Mathematics and Computer Science, TU/e. 2016-04

P.M. Heck. *Quality of Just-in-Time Requirements: Just-Enough and Just-in-Time.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2016-05

Y. Luo. *From Conceptual Models to Safety Assurance – Applying Model-Based Techniques to Support Safety Assurance.* Faculty of Mathematics and Computer Science, TU/e. 2016-06

B. Ege. *Physical Security Analysis of Embedded Devices.* Faculty of Science, Mathematics and Computer Science, RU. 2016-07

A.I. van Goethem. *Algorithms for Curved Schematization.* Faculty of Mathematics and Computer Science, TU/e. 2016-08

T. van Dijk. *Sylvan: Multi-core Decision Diagrams.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2016-09

I. David. *Run-time resource management for component-based systems.* Faculty of Mathematics and Computer Science, TU/e. 2016-10

A.C. van Hulst. *Control Synthesis using Modal Logic and Partial Bisimilarity – A Treatise Supported by Computer Verified Proofs.* Faculty of Mechanical Engineering, TU/e. 2016-11

A. Zawedde. *Modeling the Dynamics of Requirements Process Improvement.* Faculty of Mathematics and Computer Science, TU/e. 2016-12

F.M.J. van den Broek. *Mobile Communication Security.* Faculty of Science, Mathematics and Computer Science, RU. 2016-13

J.N. van Rijn. *Massively Collaborative Machine Learning.* Faculty of Mathematics and Natural Sciences, UL. 2016-14

M.J. Steindorfer. *Efficient Immutable Collections.* Faculty of Science, UvA. 2017-01

W. Ahmad. *Green Computing: Efficient Energy Management of Multiprocessor Streaming Applications via Model Checking.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2017-02

D. Guck. *Reliable Systems – Fault tree analysis via Markov reward automata.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2017-03

H.L. Salunkhe. *Modeling and Buffer Analysis of Real-time Streaming Radio Applications Scheduled on Heterogeneous Multiprocessors.* Faculty of Mathematics and Computer Science, TU/e. 2017-04

A. Krasnova. *Smart invaders of private matters: Privacy of communication on the Internet and in the Internet of Things (IoT).* Faculty of Science, Mathematics and Computer Science, RU. 2017-05

A.D. Mehrabi. *Data Structures for Analyzing Geometric Data.* Faculty of Mathematics and Computer Science, TU/e. 2017-06

D. Landman. *Reverse Engineering Source Code: Empirical Studies of Limitations and Opportunities.* Faculty of Science, UvA. 2017-07

W. Lueks. *Security and Privacy via Cryptography – Having your cake and eating it too.* Faculty of Science, Mathematics and Computer Science, RU. 2017-08

A.M. Şutîi. *Modularity and Reuse of Domain-Specific Languages: an exploration with MetaMod.* Faculty of Mathematics and Computer Science, TU/e. 2017-09

U. Tikhonova. *Engineering the Dynamic Semantics of Domain Specific Languages.* Faculty of Mathematics and Computer Science, TU/e. 2017-10

Q.W. Bouts. *Geographic Graph Construction and Visualization.* Faculty of Mathematics and Computer Science, TU/e. 2017-11

A. Amighi. *Specification and Verification of Synchronisation Classes in Java: A Practical Approach.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2018-01

S. Darabi. *Verification of Program Parallelization.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2018-02

J.R. Salamanca Tellez. *Coequations and Eilenberg-type Correspondences.* Faculty of Science, Mathematics and Computer Science, RU. 2018-03

P. Fiterău-Broştean. *Active Model Learning for the Analysis of Network Protocols.* Faculty of Science, Mathematics and Computer Science, RU. 2018-04

D. Zhang. *From Concurrent State Machines to Reliable Multi-threaded Java Code.* Faculty of Mathematics and Computer Science, TU/e. 2018-05

H. Basold. *Mixed Inductive-Coinductive Reasoning Types, Programs and Logic*. Faculty of Science, Mathematics and Computer Science, RU. 2018-06

A. Lele. *Response Modeling: Model Refinements for Timing Analysis of Runtime Scheduling in Real-time Streaming Systems*. Faculty of Mathematics and Computer Science, TU/e. 2018-07

N. Bezirgiannis. *Abstract Behavioral Specification: unifying modeling and programming*. Faculty of Mathematics and Natural Sciences, UL. 2018-08