

Toward Megamodels at Runtime

5th International Workshop on Models@run.time

Oslo, Norway, 5th October 2010

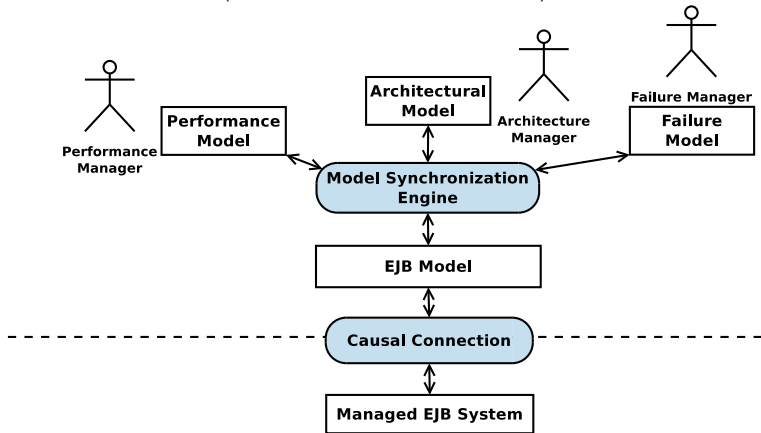
Thomas Vogel, Andreas Seibel, and Holger Giese
System Analysis and Modeling Group
Hasso Plattner Institute
University of Potsdam



Motivation

Multiple runtime models for monitoring and adaptation

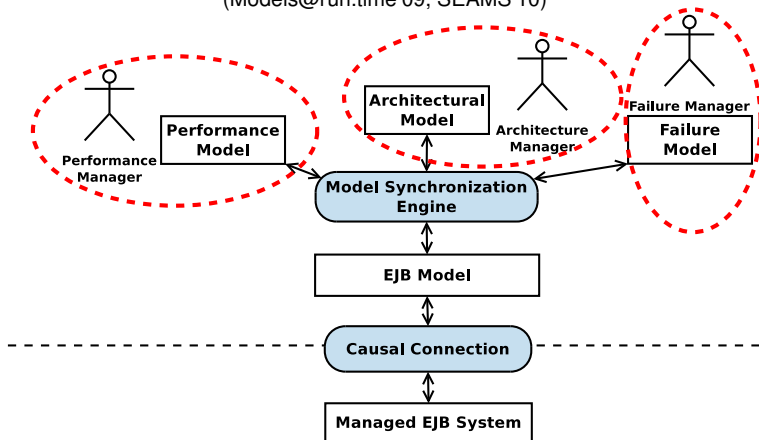
(Models@run.time'09, SEAMS'10)



Motivation

Multiple runtime models for monitoring and adaptation

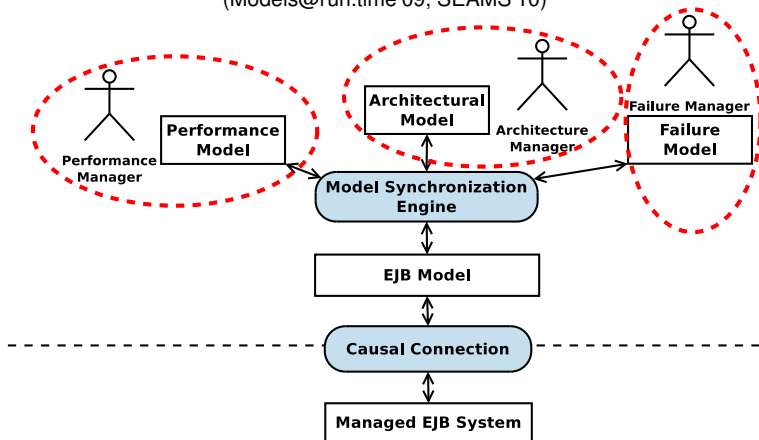
(Models@run.time'09, SEAMS'10)



Motivation

Multiple runtime models for monitoring and adaptation

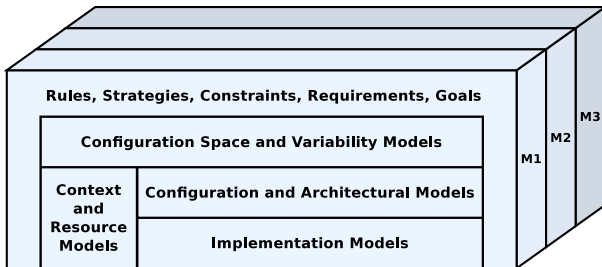
(Models@run.time'09, SEAMS'10)



Relations between models: trade-offs, dependencies, ...

Categories of Runtime Models

Simultaneous use of multiple runtime models?
Conceivable relations between runtime models?

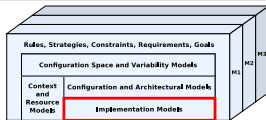


- Abstract categorization: purpose and content of a runtime model
- Based on literature, esp. the past *Models@run.time* workshops
- Categories: neither complete, nor a prerequisite for an approach

Implementation Models

Characteristics:

- Similar to models used in the field of *reflection*
- Causally connected to a running system
- Coupled to the system's implementation and computation model (solution space)



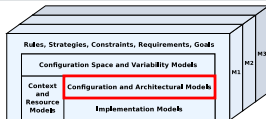
Examples:

- Reflective programming languages [JBCG06, KV08]
- Platform-specific models, like for CORBA [CPV06] or EJB [VG10]
- Class and object diagrams [JBCG06, GIWO09, Mao09]
- Sequence diagrams [Mao09]
- Statecharts, state machines, automaton [GCZ08, Mao08, HDC09]

Configuration & Architectural Models

Characteristics:

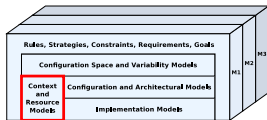
- More abstract than *Implementation Models*
- Platform-independent, problem space
- Often causally connected to a running system
- Reflect the current configuration of a system
- Software architecture as an appropriate abstraction level



Examples:

- Component diagrams, often enhanced with non-functional properties [SXC⁺10, MBJ⁺09, OMT98, GCH⁺04, VNH⁺10, VG10]
- Process or workflow models [SBVD08]
- Abstract *Implementation Models*, like statecharts for components

Context & Resource Models



Characteristics:

- Describe the system's operational environment
- Describe required or used resources (logical or physical)
- Context-aware systems

Examples:

- Some form of variables, like key value pairs [MBJ⁺09, SB08]
- Semi-structured tags and attributes, object-oriented or logic-based models [SB08]
- Feature models [ACF⁺09]

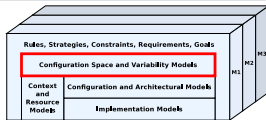
Configuration Space & Variability Models

Characteristics:

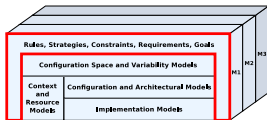
- Specify potential variants of a system
- Define the configuration space
- Used for finding adaptation options

Examples:

- Component type diagrams [GCH⁺04, GSV09, VG10]
- Feature models originating from software product lines [MBJ⁺09, CGFP09, EME09]
- Aspect models for *Configuration & Architectural Models* [MBJ⁺09, FHL⁺09]



Rules, Strategies, Constraints, Requirements, Goals



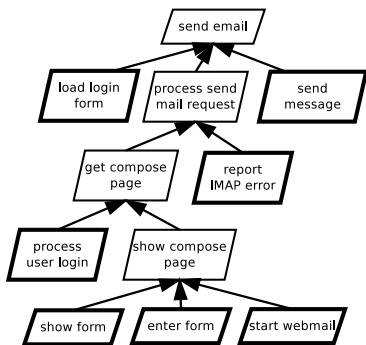
Characteristics:

- Refer to models of the other categories
- Specify adaptations (rules, strategies, goals)
- Validation and verification (constraints, requirements, goals)

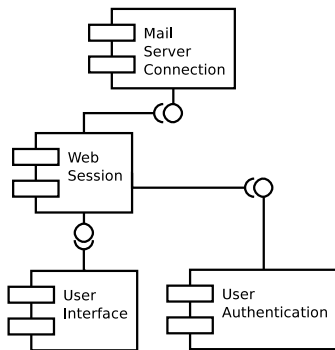
Examples:

- Event-Condition-Action (ECA) rules [GCH⁺04, ACF⁺09, DM06]
- Goal-based models (utility functions) [MBJ⁺09, EME09, RC09]
- Constraints: OCL [HRW07, VNH⁺10], Linear Temporal Logic [GCZ08]
- Goal models, like KAOS [BWS⁺10] \rightsquigarrow *requirements@run.time*

Relations between Runtime Models I

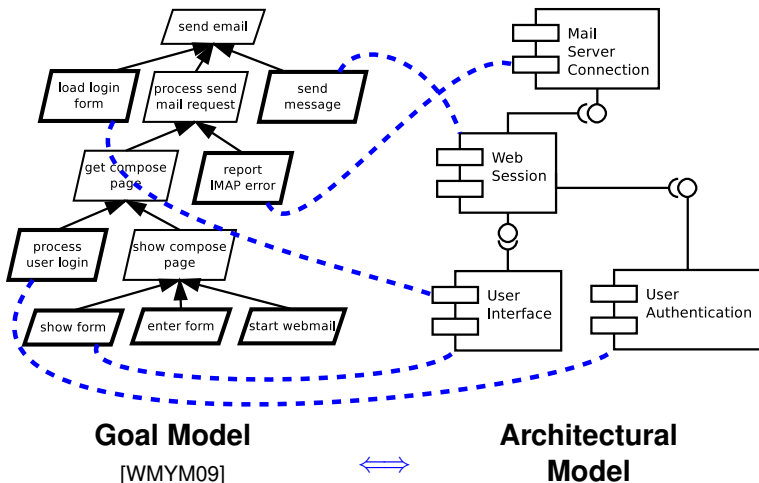


Goal Model
[WMYM09]



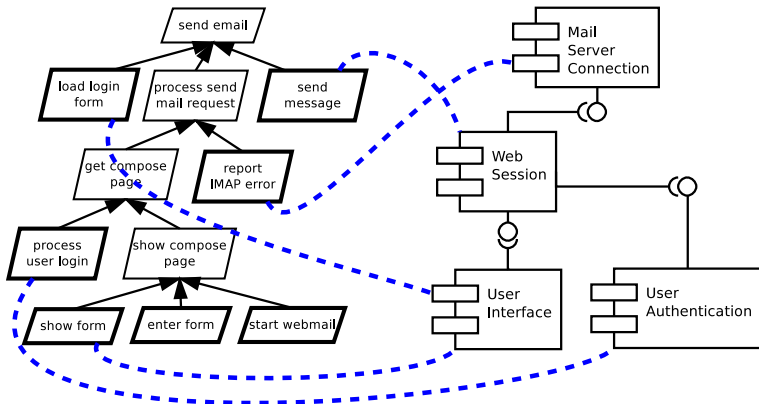
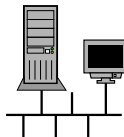
Architectural Model

Relations between Runtime Models I

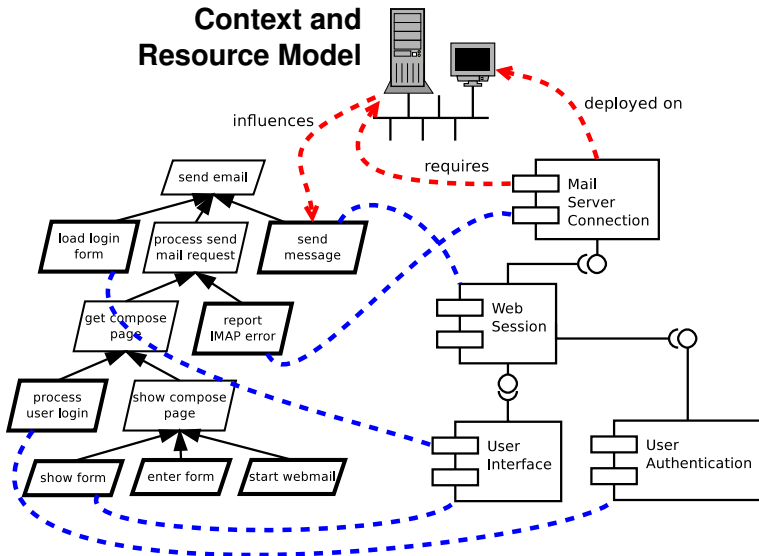


Relations between Runtime Models I

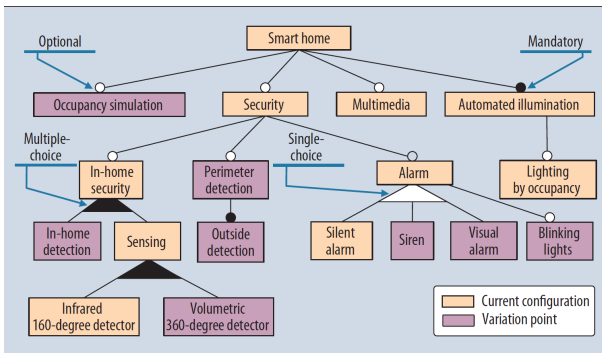
Context and Resource Model



Relations between Runtime Models I

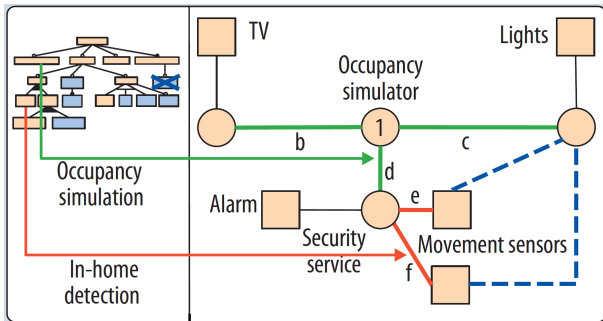


Relations between Runtime Models II



Feature Model [CGFP09]
*Configuration Space and
Variability Model*

Relations between Runtime Models II



Feature Model

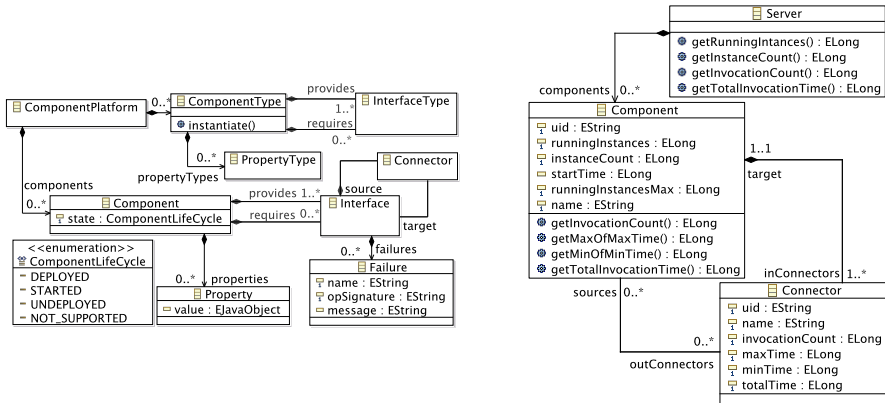
*Configuration Space
and Variability Model*

PervML Model

*Configuration and
Architectural Model*

[CGFP09]

Relations between Runtime Models IV



Failure Metamodel
Configuration and Architectural Model

↔
overlaps

Performance Metamodel
Configuration and Architectural Model

[VNH⁺10, VG10]

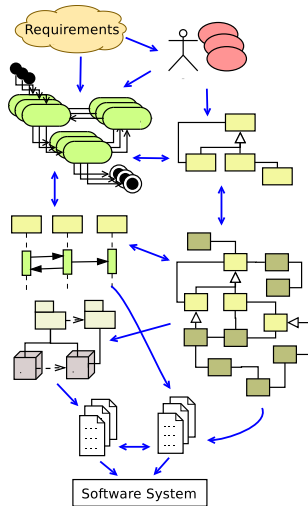
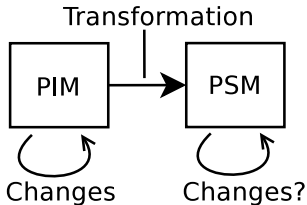
Runtime Models and Relations

- Kind of models and relations depend on the concrete approach
- It's likely that multiple models are used (vs. one model)
- Rather than isolated models, **network of runtime models**
- **Explicitly considering relations** between models
 - E.g., (impact) analysis across related models
- Existing approaches do not explicitly address these issues (ad-hoc and code-based solutions)
- Model-driven solution?

Similar Issues in MDSD

Model-Driven Software Development (MDSD)

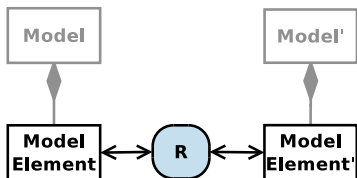
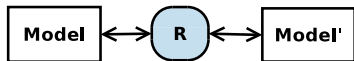
- A multitude of models and relations
- A multitude of changes
- **Consistency** among different models
- Example: Model-Driven Architecture



Megamodels

“Good enough” Definition (Megamodel)

A *megamodel* is a model that contains models and relations between those models or between elements of those models.



- Makes relations explicit
- Basis for **model-driven** management of models and relations
- Research by Favre [Fav05] and Bézivin et al. [BGMR03, BJV04, BFB07]

Megamodel Concepts

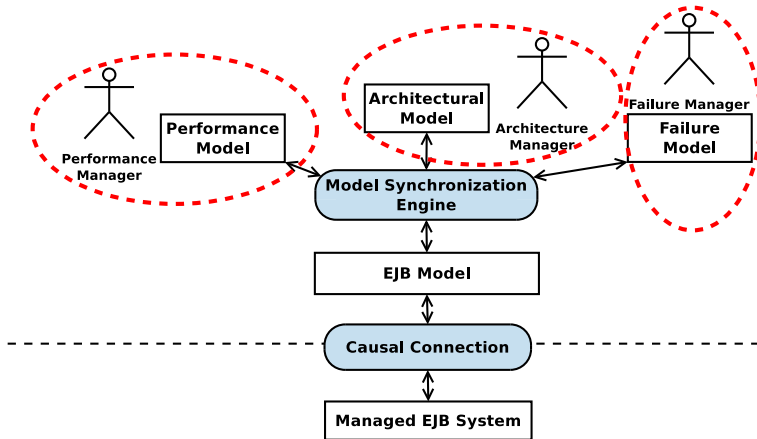
Organizational Purposes:

- Organizing and structuring models and relations
- Registry for models and their relations

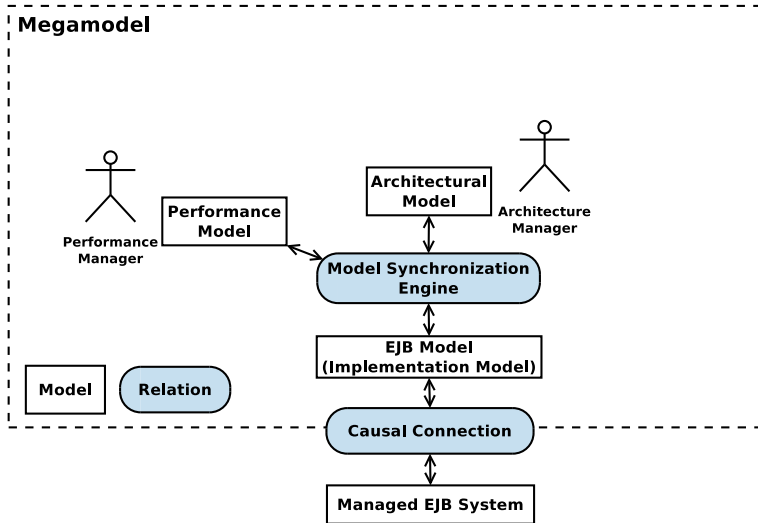
Utilization Purposes:

- Navigation through different models in a model-driven manner
- Operational relations by means of executable units

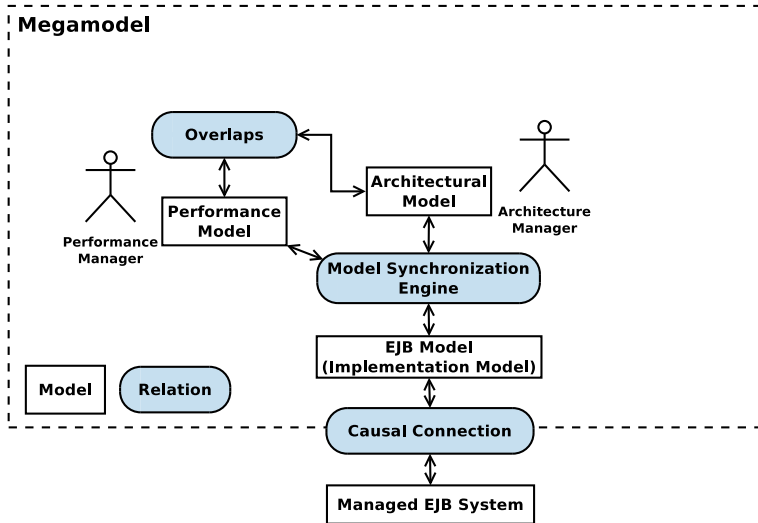
Case Study: Self-Adaptive Software



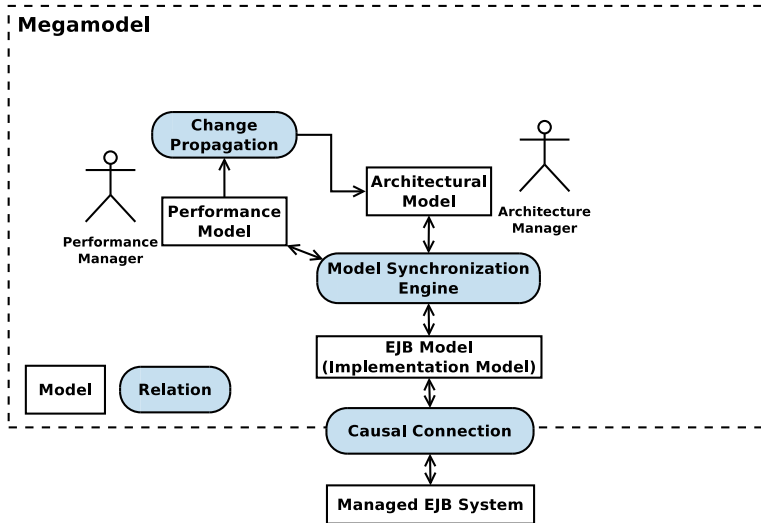
Case Study: Self-Adaptive Software



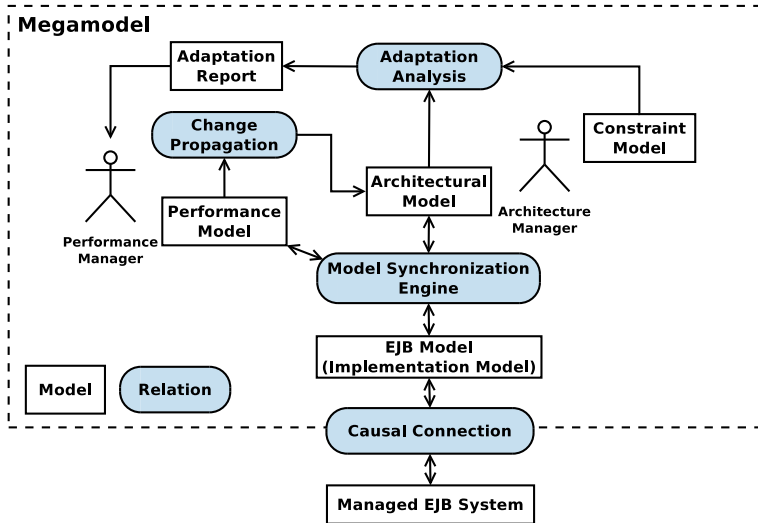
Case Study: Self-Adaptive Software



Case Study: Self-Adaptive Software



Case Study: Self-Adaptive Software



Conclusion and Future Work

Conclusion

- Multiple runtime models for advanced self-adaptive systems

⇒ **Categories of Runtime Models**

- These models are not independent from each other

⇒ **Relations between Runtime Models**

- Explicitly considering models and relations

⇒ **Megamodel concepts as a proposal**

Conclusion and Future Work

Conclusion

- Multiple runtime models for advanced self-adaptive systems

⇒ **Categories of Runtime Models**

- These models are not independent from each other

⇒ **Relations between Runtime Models**

- Explicitly considering models and relations

⇒ **Megamodel concepts as a proposal**

Future Work

- Elaborate categorization of models [FR07, Ben09, BBF09]
- Categorization of relations
- Applicability of our megamodel approach at runtime [SNG09]

References I

- [ACF⁺09] Mathieu Acher, Philippe Collet, Franck Fleurey, Philippe Lahire, Sabine Moisan, and Jean-Paul Rigault.
Modeling Context and Dynamic Adaptations with Feature Models.
In *Proc. of the 4th Intl. Workshop on Models@run.time*, volume 509 of *CEUR-WS.org*, pages 89–98, 2009.
- [BBF09] Gordon Blair, Nelly Bencomo, and Robert B. France.
Models@run.time: Guest Editors' Introduction.
Computer, 42(10):22–27, 2009.
- [Ben09] Nelly Bencomo.
On the Use of Software Models during Software Execution.
In *Proc. of the ICSE Workshop on Modeling in Software Engineering (MISE)*, pages 62–67. IEEE, 2009.
- [BFB07] Mikael Barbero, Marcos Didonet Del Fabro, and Jean Bézivin.
Traceability and Provenance Issues in Global Model Management.
In *ECMDA-TW07: Proc. of 3rd Workshop on Traceability*, pages 47–55, 2007.
- [BGMR03] Jean Bézivin, Sébastien Gérard, Pierre-Alain Muller, and Laurent Rioux.
MDA components: Challenges and Opportunities.
In *1st Intl. Workshop on Metamodelling for MDA*, pages 23–41, 2003.
- [BJV04] Jean Bézivin, Frédéric Jouault, and Patrick Valduriez.
On the Need for Megamodels.
In *Proc. of the OOPSLA/GPCE Workshop on Best Practices for Model-Driven Software Development*, 2004.
- [BWS⁺10] Nelly Bencomo, Jon Whittle, Pete Sawyer, Anthony Finkelstein, and Emmanuel Letier.
Requirements reflection: requirements as runtime entities.
In *Proc. of the 32nd ACM/IEEE Intl. Conference on Software Engineering (ICSE)*, pages 199–202. ACM, 2010.
- [CGFP09] Carlos Cetina, Pau Giner, Joan Fons, and Vicente Pelechano.
Autonomic Computing through Reuse of Variability Models at Runtime: The Case of Smart Homes.
Computer, 42(10):37–43, 2009.
- [CPV06] Fabio Costa, Lucas Provensi, and Frederico Vaz.
Towards a More Effective Coupling of Reflection and Runtime Metamodels for Middleware.
In *Proc. of 1st Intl. Workshop on Models@run.time*, 2006.
- [DM06] Jeremy Dubus and Philippe Merle.
Applying OMG D&C Specification and ECA Rules for Autonomous Distributed Component-based Systems.
In *Proc. of 1st Intl. Workshop on Models@run.time*, 2006.
- [EME09] Ahmed Elkhodary, Sam Malek, and Naeem Esfahani.
On the Role of Features in Analyzing the Architecture of Self-Adaptive Software Systems.
In *Proc. of the 4th Intl. Workshop on Models@run.time*, volume 509 of *CEUR-WS.org*, pages 41–50, 2009.
- [Fav05] Jean-Marie Favre.
Foundations of Model (Driven) (Reverse) Engineering : Models – Episode I: Stories of The Fidus Papyrus and of The Solarus.
In *Language Engineering for Model-Driven Software Development*, number 04101 in Dagstuhl Seminar Proceedings. IBFI, Schloss Dagstuhl, 2005.

References II

- [FHL⁺09] Nicolas Ferry, Vincent Hourdin, Stéphane Lavirotte, GaÅ-tan Rey, Jean-Yves Tigli, and Michel Riveill.
Models at Runtime: Service for Device Composition and Adaptation.
In *Proc. of the 4th Intl. Workshop on Models@run.time*, volume 509 of *CEUR-WS.org*, pages 51–60, 2009.
- [FR07] Robert France and Bernhard Rumpe.
Model-driven Development of Complex Software: A Research Roadmap.
In *Proc. of the ICSE Workshop on Future of Software Engineering (FOSE)*, pages 37–54. IEEE, 2007.
- [GCH⁺04] David Garlan, Shang-Wen Cheng, An-Cheng Huang, Bradley Schmerl, and Peter Steenkiste.
Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure.
Computer, 37(10):46–54, 2004.
- [GCZ08] Heather J. Goldsby, Betty H. Cheng, and Jian Zhang.
AMOEBA-RT: Run-Time Verification of Adaptive Software.
In *Models in Software Engineering: Workshops and Symposia at MoDELS 2007*, volume 5002 of *LNCS*, pages 212–224. Springer, 2008.
- [GIWO09] Tony Gjerlutsen, Mads Ingstrup, Jesper Wolff, and Olsen Olesen.
Mirrors of Meaning: Supporting Inspectable Runtime Models.
Computer, 42(10):61–68, 2009.
- [GSV09] Holger Giese, Andreas Seibel, and Thomas Vogel.
A Model-Driven Configuration Management System for Advanced IT Service Management.
In *Proc. of the 4th Intl. Workshop on Models@run.time*, volume 509 of *CEUR-WS.org*, pages 61–70, 2009.
- [HDC09] Edzard Höfig, Peter H. Deussen, and Hakan Coskun.
Statechart Interpretation on Resource Constrained Platforms: a Performance Analysis.
In *Proc. of the 4th Intl. Workshop on Models@run.time*, volume 509 of *CEUR-WS.org*, pages 99–108, 2009.
- [HRW07] Christian Hein, Tom Ritter, and Michael Wagner.
System Monitoring using Constraint Checking as part of Model Based System Management.
In *Proc. of 2nd Intl. Workshop on Models@run.time*, 2007.
- [JBCG06] Frédéric Jouault, Jean Bézivin, Régis Chevrel, and Jeff Gray.
Experiments in Run-Time Model Extraction.
In *Proc. of 1st Intl. Workshop on Models@run.time*, 2006.
- [KV08] Adrian Kuhn and Toon Verwaest.
FAME - A Polyglot Library for Metamodeling at Runtime.
In *Proc. of the 3rd Intl. Workshop on Models@run.time*, pages 57–66. Technical Report COMP-005-2008, Lancaster University, 2008.
- [Mao08] Shahar Maoz.
Model-Based Traces.
In *Proc. of the 3rd Intl. Workshop on Models@run.time*, pages 16–25. Technical Report COMP-005-2008, Lancaster University, 2008.
- [Mao09] Shahar Maoz.
Using Model-Based Traces as Runtime Models.
Computer, 42(10):28–36, 2009.

References III

- [MBJ⁺09] Brice Morin, Olivier Barais, Jean-Marc Jézéquel, Franck Fleurey, and Arnor Solberg.
Models@Run.time to Support Dynamic Adaptation.
Computer, 42(10):44–51, 2009.
- [OMT98] Peyman Oreizy, Nenad Medvidovic, and Richard N. Taylor.
Architecture-based runtime software evolution.
In Proc. of the 20th Intl. Conference on Software Engineering (ICSE), pages 177–186. IEEE, 1998.
- [RC09] Andres J. Ramirez and Betty H. Cheng.
Evolving Models at Run Time to Address Functional and Non-Functional Adaptation Requirements.
In Proc. of the 4th Intl. Workshop on Models@run.time, volume 509 of *CEUR-WS.org*, pages 31–40, 2009.
- [SB08] Daniel Schneider and Martin Becker.
Runtime Models for Self-Adaptation in the Ambient Assisted Living Domain.
In Proc. of the 3rd Intl. Workshop on Models@run.time, pages 47–56. Technical Report COMP-005-2008, Lancaster University, 2008.
- [SBVD08] Mario Sanchez, Ivan Barrero, Jorge Villalobos, and Dirk Deridder.
An Execution Platform for Extensible Runtime Models.
In Proc. of the 3rd Intl. Workshop on Models@run.time, pages 107–116. Technical Report COMP-005-2008, Lancaster University, 2008.
- [SNG09] Andreas Seibel, Stefan Neumann, and Holger Giese.
Dynamic Hierarchical Mega Models: Comprehensive Traceability and its Efficient Maintenance.
Software and Systems Modeling, 9:493–528, 2009.
- [SXC⁺10] Hui Song, Yingfei Xiong, Franck Chauvel, Gang Huang, Zhenjiang Hu, and Hong Mei.
Generating Synchronization Engines between Running Systems and Their Model-Based Views.
In Models in Software Engineering, Workshops and Symposia at MODELS 2009, volume 6002 of *LNCS*, pages 140–154. Springer, 2010.
- [VG10] Thomas Vogel and Holger Giese.
Adaptation and Abstract Runtime Models.
In Proc. of the 5th ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), pages 39–48. ACM, 2010.
- [VNH⁺09] Thomas Vogel, Stefan Neumann, Stephan Hildebrandt, Holger Giese, and Basil Becker.
Incremental Model Synchronization for Efficient Run-time Monitoring.
In Proc. of the 4th Intl. Workshop on Models@run.time, volume 509 of *CEUR-WS.org*, pages 1–10, 2009.
- [VNH⁺10] Thomas Vogel, Stefan Neumann, Stephan Hildebrandt, Holger Giese, and Basil Becker.
Incremental Model Synchronization for Efficient Run-Time Monitoring.
In Models in Software Engineering, Workshops and Symposia at MODELS 2009, volume 6002 of *LNCS*, pages 124–139. Springer, 2010.
- [WMMY09] Yiqiao Wang, Sheila A. McIlraith, Yijun Yu, and John Mylopoulos.
Monitoring and diagnosing software requirements.
Automated Software Eng., 16(1):3–35, 2009.