Models at Runtime for Adaptive and Self-managing Software

Dagstuhl Seminar on Models@run.time (2011)

Holger Giese and Thomas Vogel

System Analysis & Modeling Group
Hasso Plattner Institute for Software Systems Engineering
University of Potsdam, Germany

holger.giese@hpi.uni-potsdam.de
thomas.vogel@hpi.uni-potsdam.de

Our "definition" for Models at Runtime
Overview

- Any model used on-line to represent running software, to represent the software's environment, or to manipulate or analyze any of the former two.

- On-line: internal to the running software

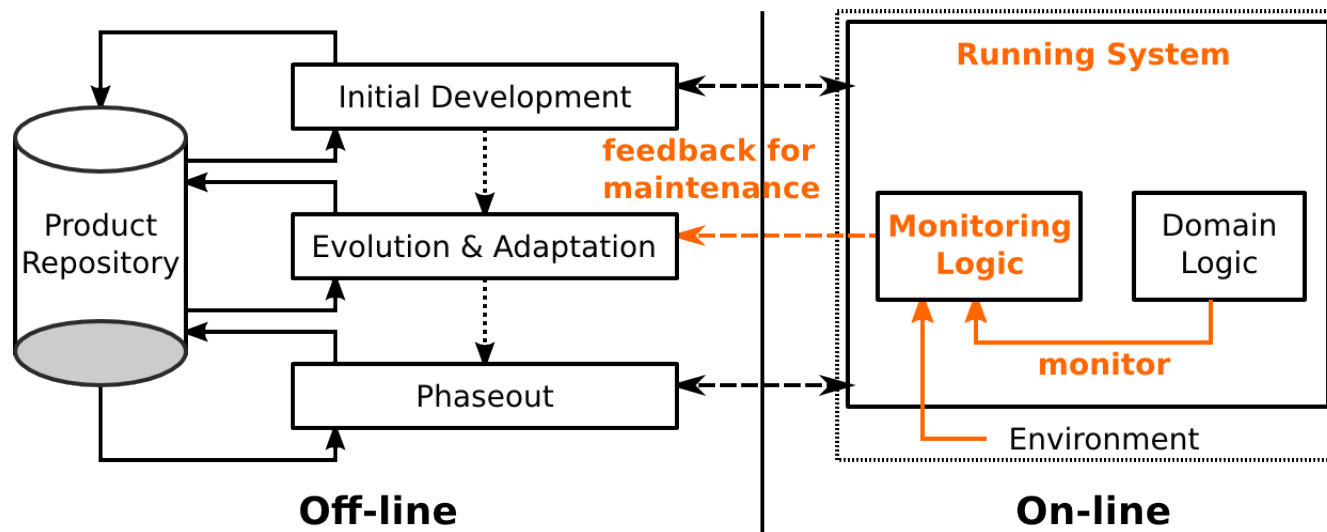- Running software: domain logic or adaptation logic of other subsystems
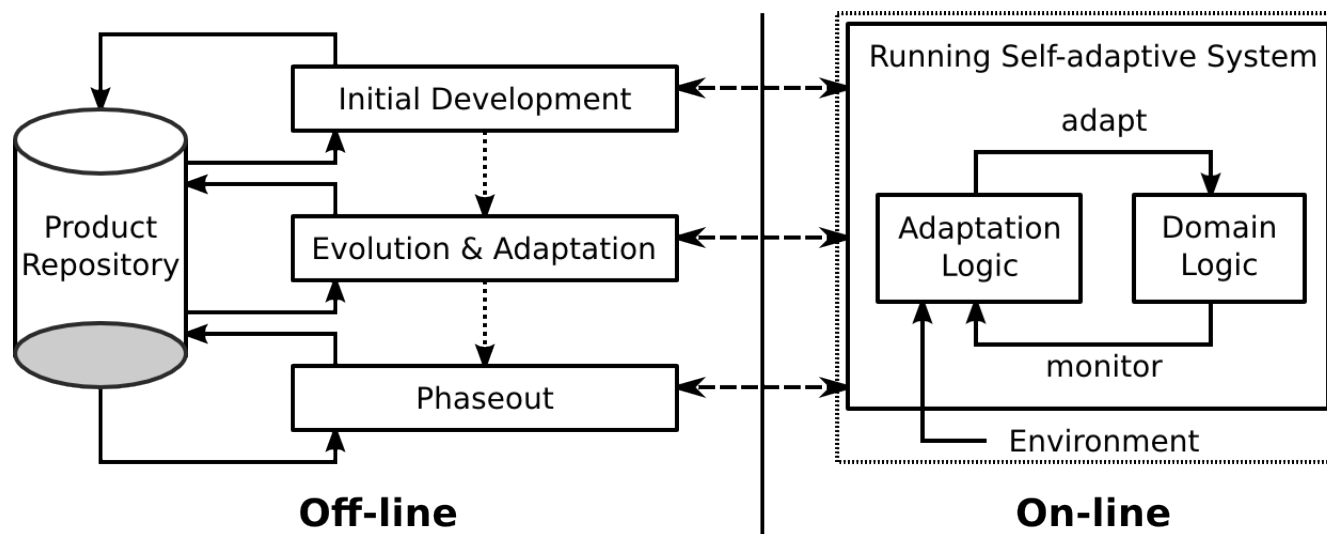
Off-line   **On-line**

Remark: On-line usage as the exclusive characteristic of "Models at Runtime"?

- Application data (model) processed by a running system as "Models at Runtime"?
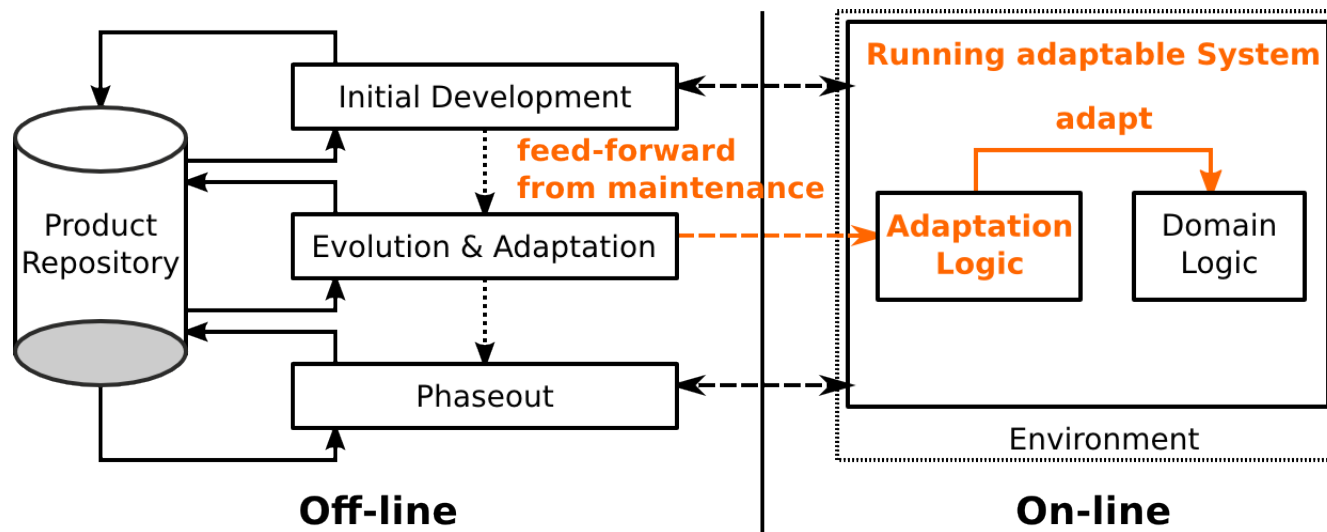
Giese & Vogel | Models at Runtime for Adaptive and Self-managing Software | 27.11. - 02.12.2011

■ Carrier of knowledge for typical (off-line) maintenance
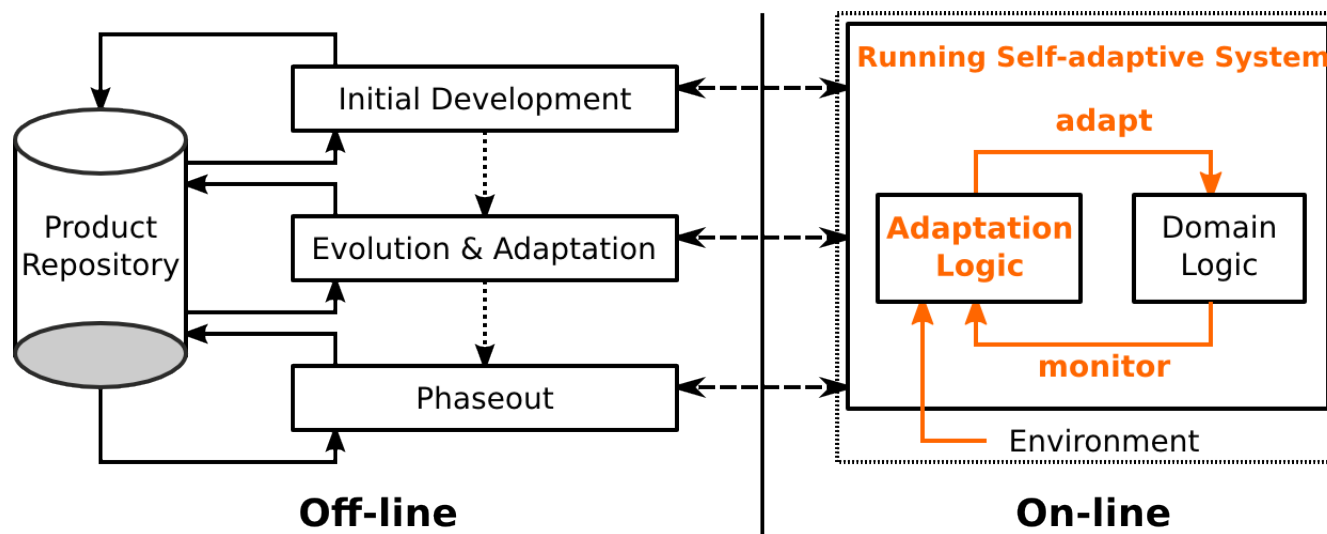
Adaptation

4

- Interface for (off-line) maintenance to change the running system

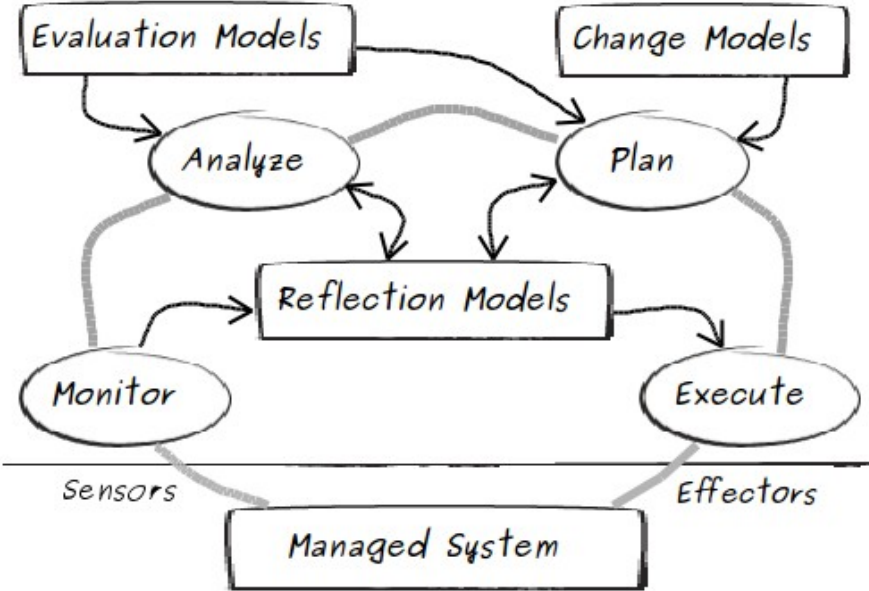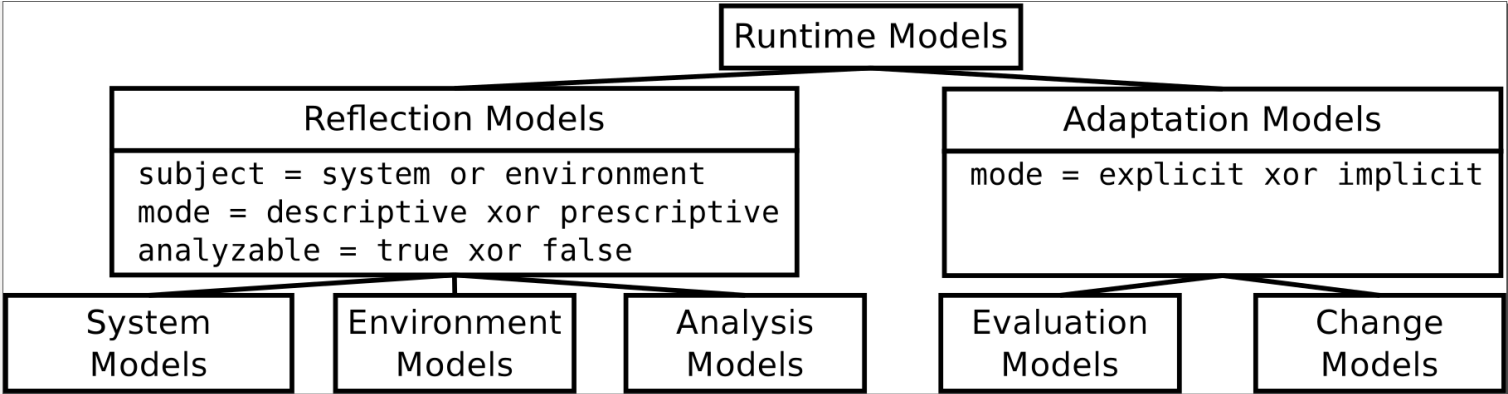# Our "definition" for Models at Runtime
Self-adaptation

- Adaptation logic implements a feedback loop, like MAPE
- **Integration** of (off-line) maintenance & (on-line) self-adaptation

Our "definition" for Models at Runtime
## Categories of Runtime Models

6

- APIs vs. Runtime Models
  - □ (Sensors & Effectors)
- Not necessarily MDE (meta- and meta-metamodel levels) for all Runtime Models
- … but promising!?
  - □ Abstraction
  - □ Automation
  - □ Relation to dev-time models

- Formal foundations based on graph transformations

- Multiple reflection models and adaptation loops (different concerns and abstractions)

- Incremental model synchronization techniques

  □ Based on triple graph grammars (graph transformations)

- Adaptation models: executable, interpreted models

  □ Based on Story Diagrams (graph transformations)

  □ Based on Story Patterns (incremental graph transformations)

- **Megamodel: specifying an adaptation loop**

- **Megamodels: abstractions of adaptation loops in systems with multiple loops**

Megamodel:
Specifying an Adaptation loop
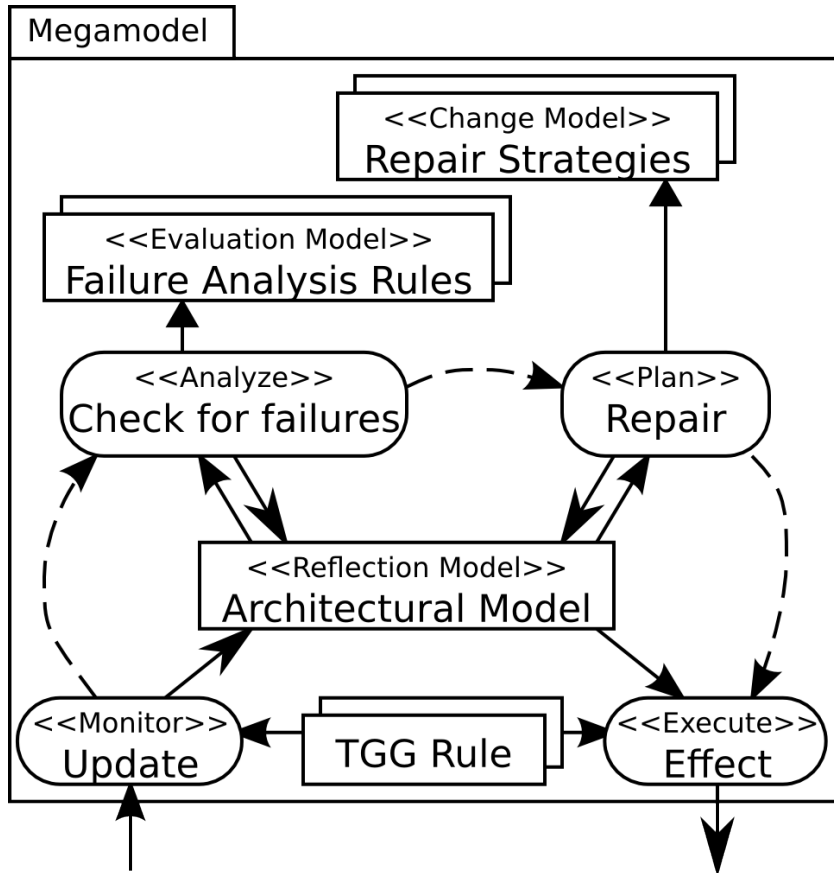
- MAPE as model operations on runtime models
- Megamodels
  - Specification of a loop
  - Structuring models and operations
  - Operationalization for MAPE
- Story Diagrams/Patterns + OCL for Evaluation and Change Models
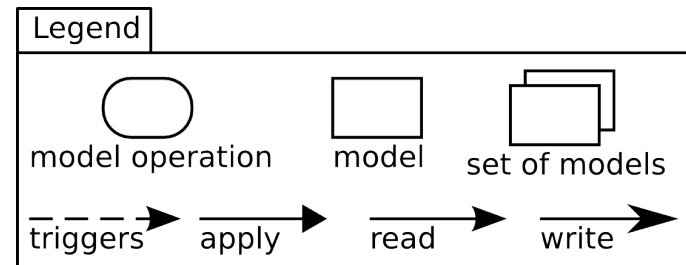
Megamodels:
Abstractions of Adaptation Loops in Systems with Multiple Loops
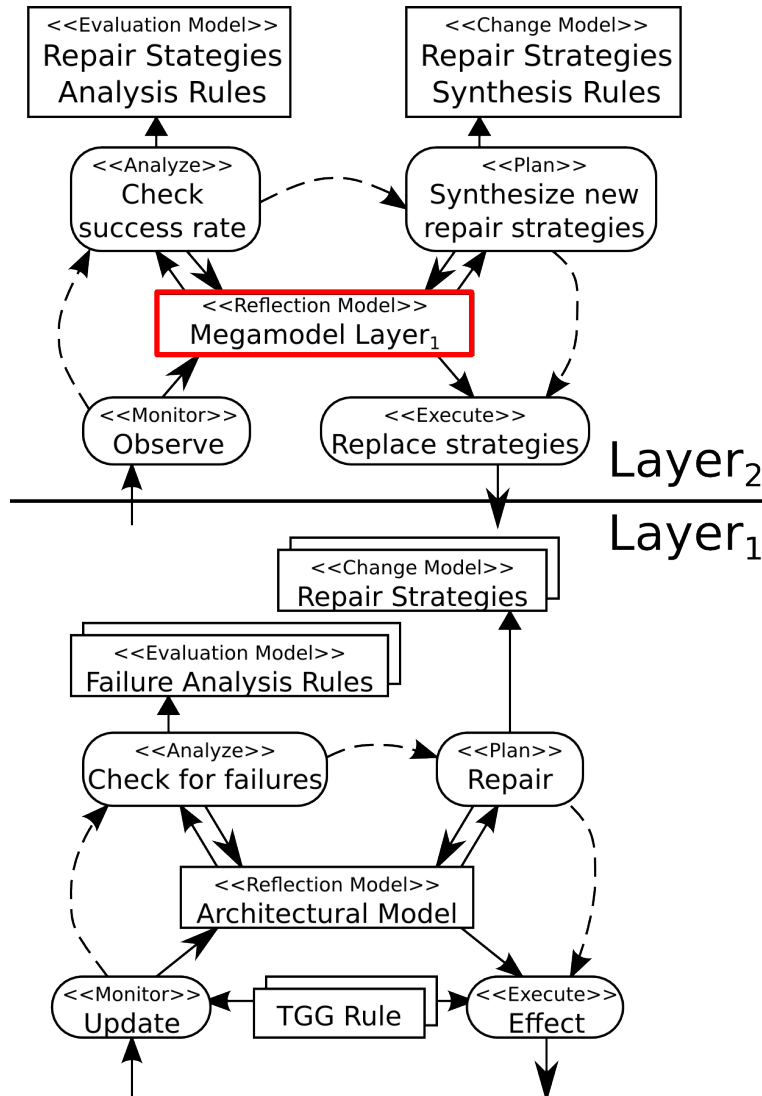
- Layered architectures for self-managing software systems (cf. Kramer & Magee, 2007)

- Each layer as a loop

- Timely decoupled layers:

  - □ $Layer_0$ – Running system

  - □ $Layer_1$ – Architectural Reflection

  - □ $Layer_2$ – Megamodel Reflection

- Specialized vs. generic MDE/SE techniques for the runtime, and their trade-offs, e.g., concerning efficiency and development costs

- **Uncertainty** concerning the running system and its environment

  - Incomplete or imprecise information due to
    - Abstraction
    - Adaptive (not continuous) monitoring
    - Unobservable phenomena
    - Measurement errors
    - ...

  - How models at runtime may help?

- **Assurance** for self-adaptive software concerning

  - the core functionality (domain logic) in case of models at runtime
  - the adaptation loops (adaptation logic) based on models at runtime

Layered Architecture:

- Jeff Kramer and Jeff Magee. Self-Managed Systems: an Architectural Challenge. In FOSE'07: 2007 Future of Software Engineering, Pages 259-268, IEEE Computer Society, Washington, DC, USA, 2007.

Feedback/Control/Adaptation Loop

- Jeffrey O. Kephart and David Chess. The Vision of Autonomic Computing. In Computer, Vol. 36(1):41--50, IEEE Computer Society Press, Los Alamitos, CA, USA, 2003.
- Yuriy Brun, Giovanna Di Marzo Serugendo, Cristina Gacek, Holger Giese, Holger M. Kienle, Marin Litoiu, Hausi A. Müller, Mauro Pezzè and Mary Shaw. Engineering Self-Adaptive Systems through Feedback Loops. In Betty Cheng, Rogerio Lemos, Holger Giese, Paola Inverardi and Jeff Magee (editors), Software Engineering for Self-Adaptive Systems, Vol. 5525:48-70 of LNCS, Springer, 2009.

Own publications related to Models@run.time:

- Thomas Vogel and Holger Giese. Language and Framework Requirements for Adaptation Models. In Proc. of the 6th International Workshop on Models@run.time, Wellington, New Zealand, Vol. 794:1-12 of CEUR-WS.org, 2011.
- Thomas Vogel, Andreas Seibel and Holger Giese. The Role of Models and Megamodels at Runtime. In Models in Software Engineering, Workshops and Symposia at MODELS 2010, Oslo, Norway, October 3-8, 2010, Reports and Revised Selected Papers, Vol. 6627:224-238 of LNCS, Springer-Verlag, 2011.
- Thomas Vogel and Holger Giese. Adaptation and Abstract Runtime Models. In Proc. of the 5th Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2010), Cape Town, South Africa, Pages 39-48, ACM, 2010.
- 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, Denver, CO, USA, October 4-9, 2009, Reports and Revised Selected Papers, Vol. 6002:124-139 of LNCS, Springer-Verlag, 2010.
- Holger Giese, Andreas Seibel and Thomas Vogel. A Model-Driven Configuration Management System for Advanced IT Service Management. In Proc. of the 4th International Workshop on Models@run.time, Denver, Colorado, USA, Vol. 509:61-70 of CEUR-WS.org, 2009.
- Thomas Vogel, Stefan Neumann, Stephan Hildebrandt, Holger Giese and Basil Becker. Model-Driven Architectural Monitoring and Adaptation for Autonomic Systems. In Proc. of the 6th IEEE/ACM International Conference on Autonomic Computing and Communications (ICAC 2009), Barcelona, Spain, Pages 67-68, ACM, 2009.