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

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

- Application data (model) processed by a running system as “Models at Runtime”?
Our “definition” for Models at Runtime Monitoring

- Carrier of knowledge for typical (off-line) maintenance
Our “definition” for Models at Runtime Adaptation

- 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

- Reflection Models
  - subject = system or environment
  - mode = descriptive xor prescriptive
  - analyzable = true xor false

- Adaptation Models
  - mode = explicit xor implicit

- System Models
- Environment Models
- Analysis Models
- Evaluation Models
- Change Models

- 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
Our research: Models at Runtime for Self-adaptive Software Systems

- 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

- Each layer as a loop
- Timely decoupled layers:
  - Layer₀ – Running system
  - Layer₁ – Architectural Reflection
  - Layer₂ – Megamodel Reflection
Topics and Challenges

- 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
References and Further Reading

Layered Architecture:

Feedback/Control/Adaptation Loop

Own publications related to Models@run.time: