PCM & Runtime Adaptation
Motivation

Elastic System Design: How to model it?
Modelling…?

- How do we represent the system, i.e., what is its meta-model?
  - Design-Time Structure: Types, potential connectors, potential deployments, …
  - Initial State: Initial component instances, connectors, deployment
  - Runtime State: Current configuration, component instances, actual deployments, *environment sensor data*

- How do we model the system changes?
  One solution: graph transformation systems (GTS)

- How do we check properties of these systems?
  One solution: graph-based model checking
Views and Viewpoints

- When defining a new modelling language
  - define viewpoint
  - and views

- Viewpoint
  - Modelled system aspect
  - Set of views for a specific purpose

- Views
  - Model system parts with specific focus
  - Defines notation and filter criteria
Views and Viewpoints

Static Viewpoint

• Potential component structure, types
• Monitoring probe locations

Adaptation Viewpoint

• Initial state view
• Adaptation rules view
**System Type View + Monitoring Specification**

- **LoadBalancerNode**
  - Load Balancer
  - <<monitoring>> ArrivalRate
- **ServerNode[i]**
  - Server[i]
  - i:1..*
- **Utilisation[i]**
- **Utilisation[i]**
  - <<leastUtil>>
Initial State View

- **Ibn:** LoadBalancerNode
  - **Ib:** Load Balancer

- **Sn[1]:** ServerNode[i]
  - **S[1]:** Server[i]
Adaptation View

\[ \exists i: \text{Utilisation}[i] > \text{ut} \]

<<condition>>

<<timing>>

\[ \text{clock} < 10 \text{s} \]
Adaptation Views

- Adaptation views are graph transformations
  - transform the Model@Runtime
  - span a graph language together with the initial state graph
- Special formalisms and tools exist
  - GTS
  - Story Charts (GTS with control flow)
  - Timed Story Charts
  - Prob. Timed Story Charts
- State space checker
Self-Healing Rules – Possible Parts

Context (matching the runtime model)

Trigger Condition (often refers to the monitoring model)

Context

<<assumption>>
Application Delay = 1 sec

Corrective Action (updating runtime model and system)

<<expected>>
Utilisation_CPU < 50%
Deadline = 10 sec

Assumption in the “nothing changes” case
Control Loop

- Apply Matching Graphreplacement
- Update Monitoring
- Predict Change Impact
- Apply Matching Graphreplacement

[Trigger Formula violated]

[not ok] / rollback

[ok] / rollback

S. Becker, Modelling CB Systems @ Runtime
**Conclusion**

Adaptive systems need new (architectural?) viewtypes

Static views is extended by additional information
- Cardinalities
- Connector types
- Monitoring locations

Adaptation views are a kind of graph grammar
- In-place model transformations, i.e., GTS
- Initial state

Future work
- Use such models to implement, predict, model-check (?), and manage systems at run-time