Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Architectural Models at Runtime

Talk at ArchiPL 2009, Munich, 11/2009

Holger Giese
System Analysis & Modeling Group
Hasso Plattner Institute for Software Systems Engineering
at the University of Potsdam, Germany
holger.giese@hpi.uni-potsdam.de

Hypothesis

Linking the Architectural Model(s) and the Programming Language is not enough? You must also link them to the Runtime System!

Why Runtime Representation?

Autonomic Computing / Self-Managed / Self-Adaptive Software:

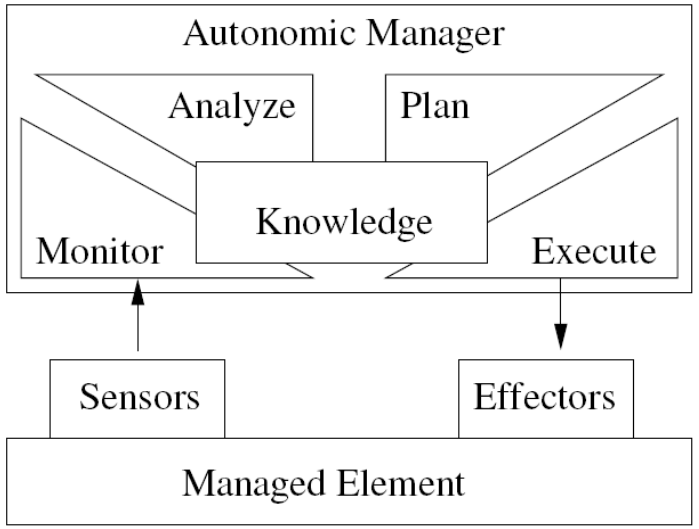
- Sensors and effectors usually at the abstraction level of APIs

⇨ Self-Management limited to simple parameters

⇨ The position of sensor/effectors in the architecture are not captured

How can we do self-management also for dynamic architectures?

- Architectural views on a managed elements is required?

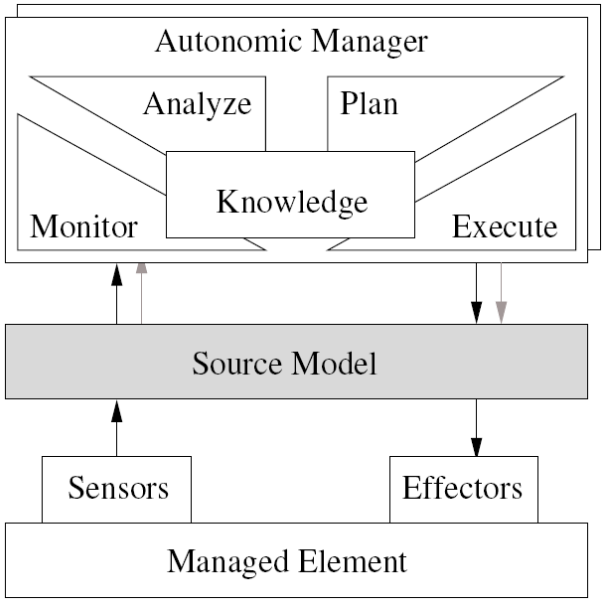


[Kephart and Chess, 2003]

⇨ Runtime representation (model) of the architecture is required

One Architectural Model at Runtime

- Runtime model of a managed element: **Source Model**
- Model-based sensors and effectors for monitoring and adaptation

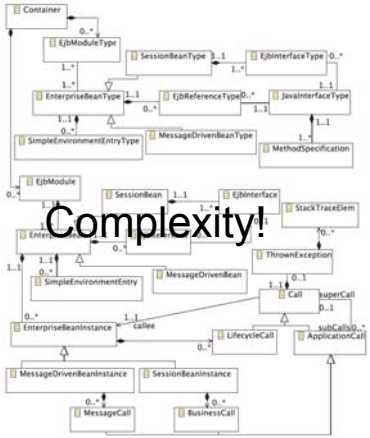


Problem:

- One **complex** model: types; deployed components and their configurations; concrete instances and interactions

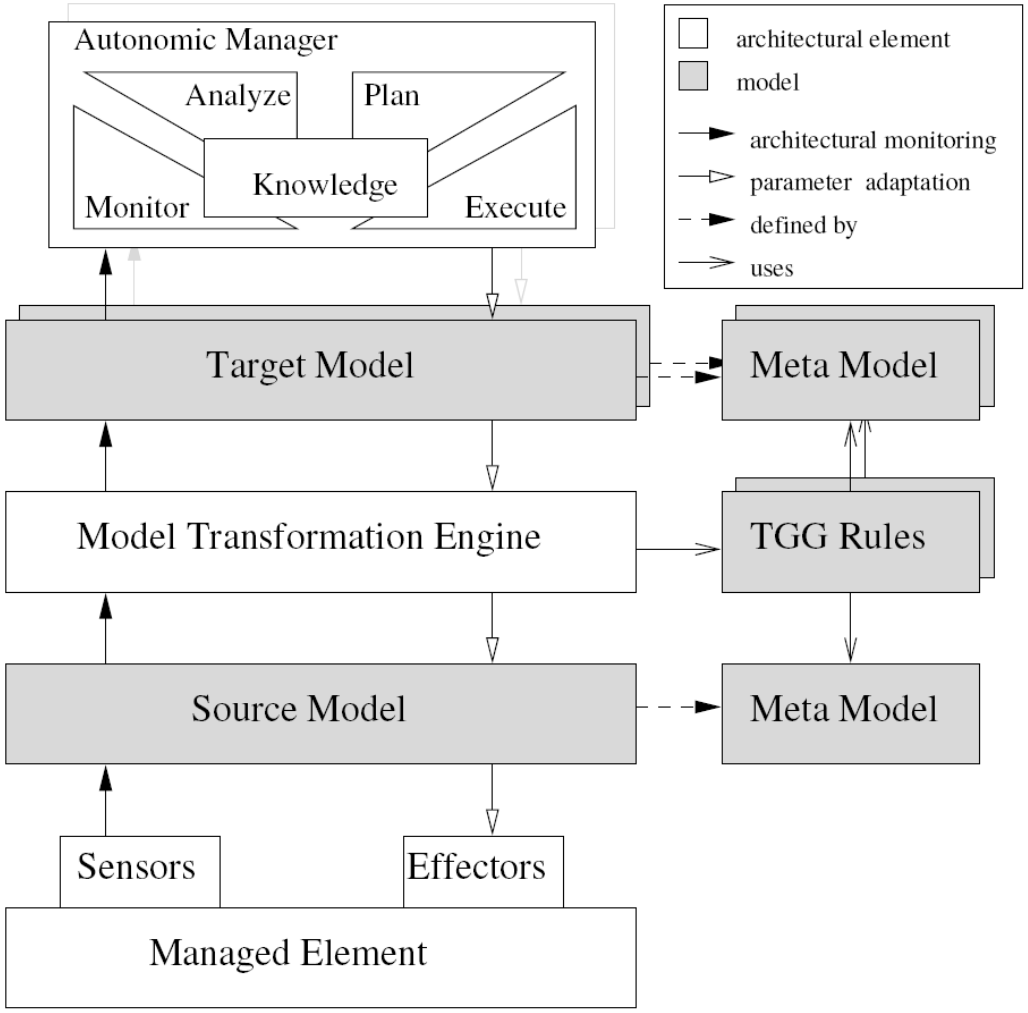
Observation:

- Each self-* capability only requires its specific architectural view, but the autonomic managers have to cope with the whole application specific complexity



Multiple Architectural Models at Runtime

- **Target models**: higher level of abstraction and a specific view on a managed element
- Maintenance of target models by incremental and bidirectional model synchronization based on **Triple Graph Grammars (TGG)**
- Changes of the source model are reflected incrementally in target models (**monitoring**) and vice versa (**adaptation**)



[Vogel+2009a,Vogel+2009b]

Experiment with EJB and 3 Views

EJB Prototype

- EJB Glassfish application server + mKernel extension for monitoring
- Source models conform to EMF and is updated event-driven

View 1: Architectural constraints for self-configuration

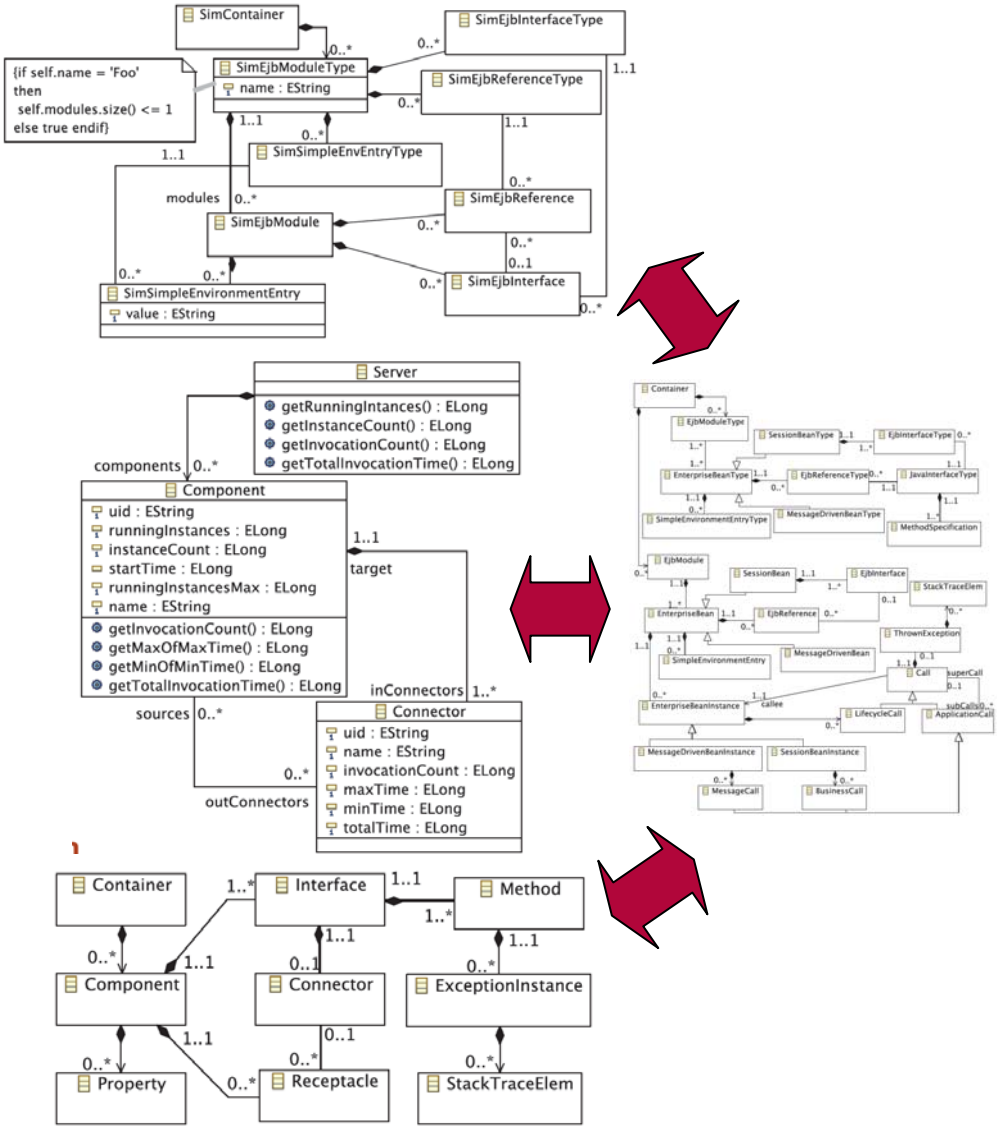
- Simplified runtime architectures of EJB-based applications for checking architectural constraints using OCL

View 2: Performance data for self-optimization

- Architectural information enriched with performance data

View 3: Failure data for self-healing

- Architectural information enriched with occurred failures



Efficient enough?

Size	Model-Driven Approach						Non-inc.
	n=0	n=1	n=2	n=3	n=4	n=5	
5	0	163	361	523	749	891	8037
10	0	152	272	457	585	790	9663
15	0	157	308	472	643	848	10811
20	0	170	325	481	623	820	12257
25	0	178	339	523	708	850	15311
Event processing	0%	92.8%	94.1%	95.6%	95.2%	96.3%	-
Synchronization	0%	7.2%	5.9%	4.4%	4.8%	3.7%	-

[ms]

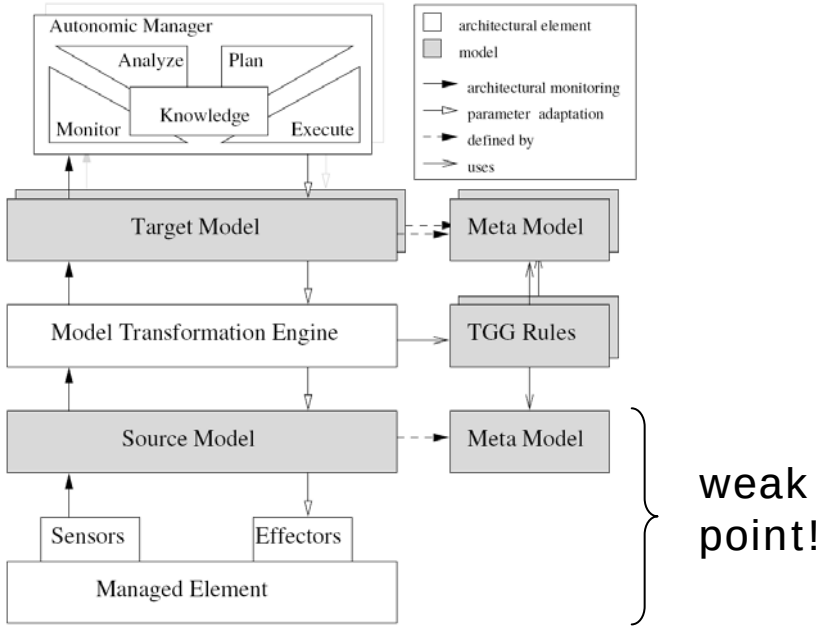
< 30 ms

Performance:

- Size: number of deployed beans
- Processing n events and invoking once the transformation engine
- **Problem:** glue code to mKernel results in enormous overhead

⇨ direct support for architectural representation would boost performance

Conclusion & Implication



weak point!

Future Work / Open Points

- Architectural adaptations (even in case of abstraction)
- Dependencies among views
- Coordinate autonomic managers
- Distributed setting

Observations

- Similar views are also required for **software maintenance** activities
- The Source model is **platform specific** while target models are often **platform independent** (reuse!)
- Generic EJB + mKernel based approach results in unnecessary overhead

Implication

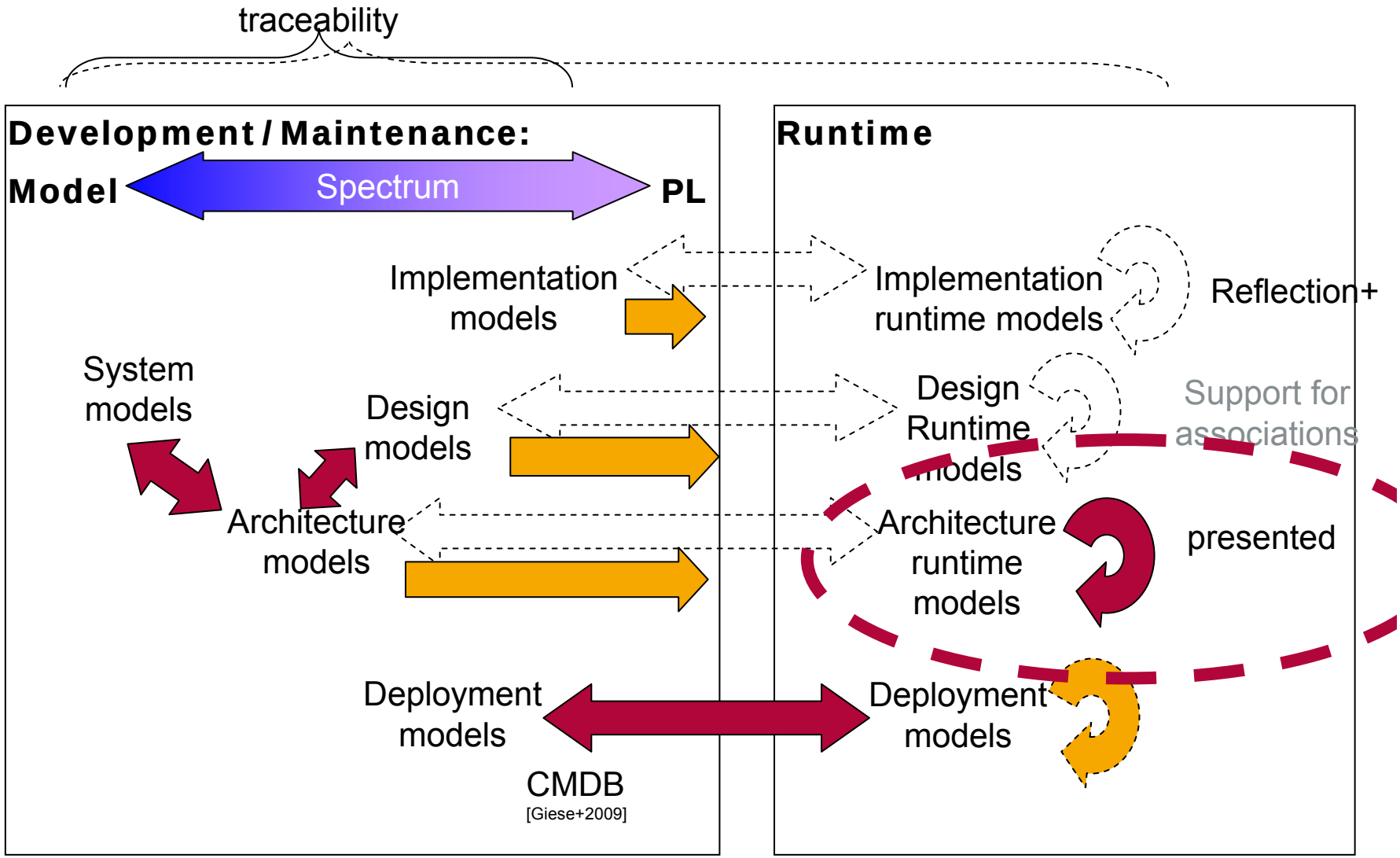
- Efficient support for architectural information at runtime is required!

Options:

a) Generating application specific code that directly provides an interface in form of a source model

b) Programming language support for architectural concepts and efficient reflection (with **write** capabilities)

The Big Picture



Published Work and References

[Vogel+2009a] Thomas Vogel and Stefan Neumann and Stephan Hildebrandt and Holger Giese and Basil Becker. Model-Driven Architectural Monitoring and Adaptation for Autonomic Systems- In Proceedings of the 6th International Conference on Autonomic Computing and Communications (ICAC 2009), Barcelona, Spain, ACM, 6 2009, p. 67-68.

[Giese+2009] Holger Giese and Andreas Seibel and Thomas Vogel. A Model-Driven Configuration Management System for Advanced IT Service Management. In Proceedings of the 4th International Workshop on Models@run.time at the 12th IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS 2009), Denver, Colorado, USA, (Nelly Bencomo and Gordon Blair and Robert France and Cedric Jeanneret and Freddy Munoz, ed.), vol. 509, CEUR Workshop Proceedings, 10 2009, p. 61-70.

[Vogel+2009b] Thomas Vogel and Stefan Neumann and Stephan Hildebrandt and Holger Giese and Basil Becker. Incremental Model Synchronization for Efficient Run-time Monitoring. In Proceedings of the 4th International Workshop on Models@run.time at the 12th IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS 2009), Denver, Colorado, USA, (Nelly Bencomo and Gordon Blair and Robert France and Cedric Jeanneret and Freddy Munoz, ed.), vol. 509, CEUR Workshop Proceedings, 10 2009, p. 1-10.

[Kephart and Chess, 2003] Kephart, J. O. and Chess, D. M.: The Vision of Autonomic Computing. IEEE Computer, 36(1):41–50, (2003).