

Runtime Models as Interfaces for Adapting Software Systems

Seminar on Software Engineering for Self-Adaptive Systems

Schloss Dagstuhl, October 24-29, 2010

Thomas Vogel
System Analysis and Modeling Group
Hasso Plattner Institute
University of Potsdam



Introduction

*The term Model-Driven Engineering (MDE) is **typically** used to describe software **development** approaches in which abstract models of software systems are created and systematically transformed to concrete implementations.*

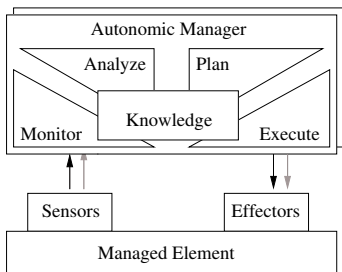
...

*In our broad vision of MDE, models are not only the primary artifacts of development, they are also the **primary means by which developers and other systems understand, interact with, configure and modify the runtime behavior of software.***

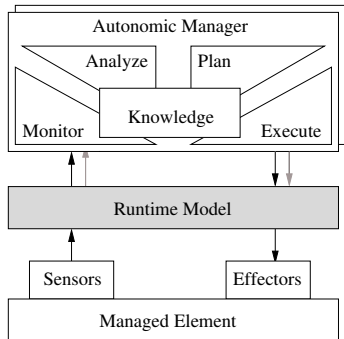
[France and Rumpe, 2007]

- Runtime models as interfaces for adaptation
- Typical or new MDE concepts and techniques for self-adaptive software systems (online vs. offline)

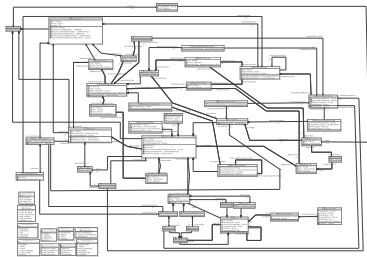
One Runtime Model as an Interface



[Kephart and Chess, 2003]



Abstract Runtime Models



complex

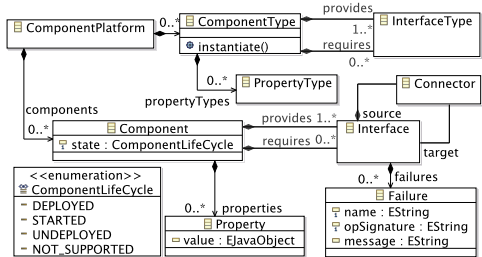
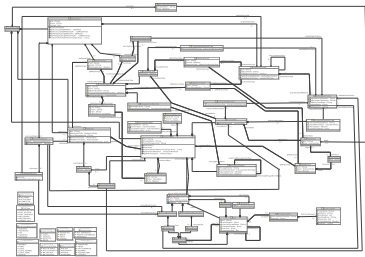
detailed

multiple concerns

platform-specific

solution space

Abstract Runtime Models



complex

detailed

multiple concerns

platform-specific

solution space

VS.

less complex

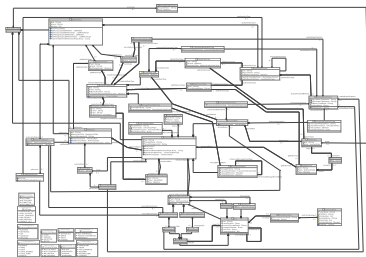
abstract

one concern

platform-independent

problem space

Abstract Runtime Models

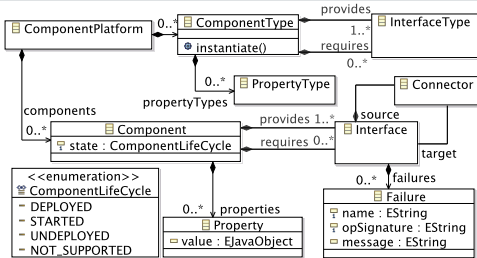


*complex
detailed*

*multiple concerns
platform-specific
solution space*

**Metamodel for a
Source Model**

VS.



*less complex
abstract*

*one concern
platform-independent
problem space*

**Metamodel for a
Target Model**

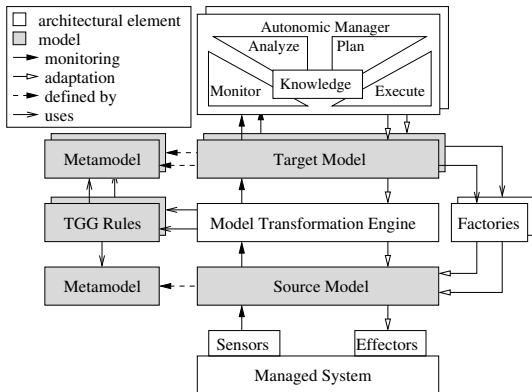
MDE for Self-Adaptive Software Systems

Different runtime models for **monitoring**

- performance,
- failures, and
- architectural constraints,

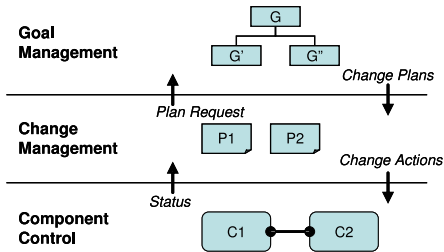
and for parameter and structural **adaptation**.

Incremental, bidirectional model synchronization based on **Triple Graph Grammars** (TGG).

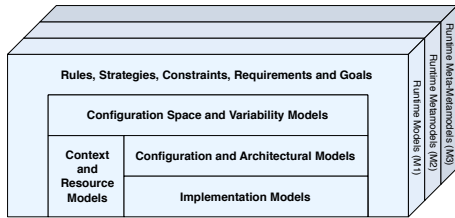


Ongoing and Future Work

Which kind of models at which level of abstraction for which adaptation/management task? \rightsquigarrow Architecting self-adaptive systems



[Kramer and Magee, 2007]



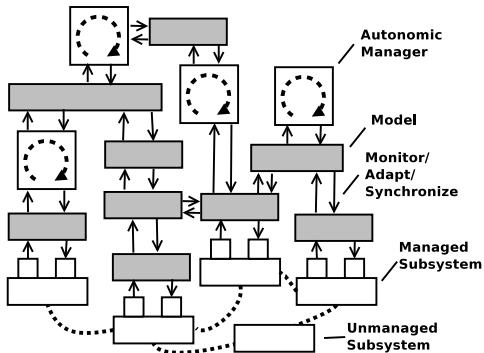
[Vogel et al., 2010b]

Using models as interfaces for managed systems and within autonomic managers, how to specify the **semantics** of models and model operations?

Ongoing and Future Work cont'd

Distributed Self-Adaptive Systems \rightsquigarrow Self-organizing systems

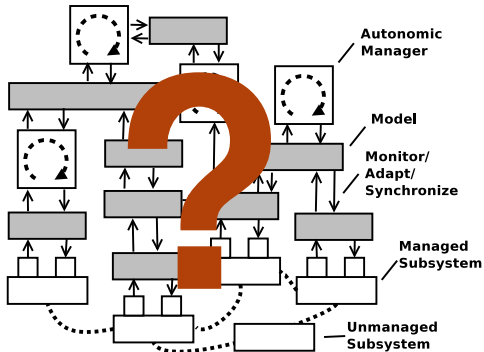
- Distributed managed *and* managing systems
- Distribution vs. consistency
- **Distributed models and MDE techniques**
- Specialized models for
 - monitoring and adapting managed systems
 - coordination between autonomic managers
 - ...



Ongoing and Future Work cont'd

Distributed Self-Adaptive Systems \rightsquigarrow Self-organizing systems

- Distributed managed *and* managing systems
- Distribution vs. consistency
- **Distributed models and MDE techniques**
- Specialized models for
 - monitoring and adapting managed systems
 - coordination between autonomic managers
 - ...



References

- [France and Rumpe, 2007] France, R. and Rumpe, B. (2007).
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.
- [Kephart and Chess, 2003] Kephart, J. and Chess, D. (2003).
The Vision of Autonomic Computing.
IEEE Computer, 36(1):41–50.
- [Kramer and Magee, 2007] Kramer, J. and Magee, J. (2007).
Self-Managed Systems: an Architectural Challenge.
In *Proc. of the ICSE Workshop on Future of Software Engineering*, pages 259–268. IEEE.
- [Vogel and Giese, 2010] Vogel, T. and Giese, H. (2010).
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.
- [Vogel et al., 2009] Vogel, T., Neumann, S., Hildebrandt, S., Giese, H., and Becker, B. (2009).
Model-Driven Architectural Monitoring and Adaptation for Autonomic Systems.
In *Proc. of the 6th Intl. Conference on Autonomic Computing and Communications*, pages 67–68. ACM.
- [Vogel et al., 2010a] Vogel, T., Neumann, S., Hildebrandt, S., Giese, H., and Becker, B. (2010a).
Incremental Model Synchronization for Efficient Run-Time Monitoring .
In *Models in Software Engineering, Workshops and Symposia at MODELS 2009, Reports and Revised Selected Papers*, volume 6002 of LNCS, pages 124–139. Springer.
- [Vogel et al., 2010b] Vogel, T., Seibel, A., and Giese, H. (2010b).
Toward Megamodels at Runtime.
In *Proceedings of the 5th International MODELS Workshop on Models@run.time*, volume 641 of *CEUR Workshop Proceedings*, pages 13–24. CEUR-WS.org.