Evolution of the Palladio Component Model: Process and Modeling Methods

SOSP 2014 - Stuttgart

Reiner Jung[1] Philipp Merkle[2] Misha Strittmatter[2]

[1] Kiel University
[2] Karlsruhe Institute of Technology

28th November 2014

iObserve

This work was partially supported by the DFG (German Research Foundation) under the Priority Program SPP1593: Design For Future - Managed Software Evolution (grant HA 2038/4-1, RE 1674/6-1, PO 607/3-1) and the Helmholtz Association of German Research Centers

Motivation

Current state PCM [BKR09] (approx. over 10 years old)

* Large, monolithic meta-model
* Different, incompatible variations
* Design smells in meta-model syntax and semantics
* Inconsistent naming schemes

Challenge Evolution of PCM into an extensible and modular meta-model

Solution

* Applying design principles for meta-models
* Implementation plan of the PCM evolution

Sample Issues of the PCM

Structure

* Large meta-models (e.g., 147 classes pcm.ecore)
* Type and instance modeling mixed together

References

* Containment references 89
* Many explicit container references (85)
  * **pcm.core.PCMRandomVariable** (17)
* Reference names, like **referenceName_DefiningClassName** (5 different styles)

Element Names

* Realized in **pcm.core.entity.NamedElement** with **entityName** property and not **name**
* Not compulsory and unique in context
* Used with and without package naming

Related Solution Ideas

cf. [Jun+14]

Discussed Solutions

* Classic EMF extension approach [Ste+09]
* Decorator pattern approach [Gam+94; Str+13]
* EMF profiles/PCM profiles [Lan+12; Kra+12]

Shortcomings

* Limited extendability
* Additional dependencies or complex meta-models
* EMF profiles are not meta-models
* Incompatible with (some) EMF tooling (e.g., GenModel, Xtext, ATL)
* Missing plan to actually evolve PCM

Solution Overview

Based on our VAO 2014 workshop paper [Jun+14]

Modularization Approach

* Basics of Meta-Models
* Base and Aspect Meta-Models Separation
* Use-Cases based Meta-Model Pattern

Evolution Approach

* Stakeholders and other Obstacles
* Evolution Plan

Basics of Meta-Models

Elements

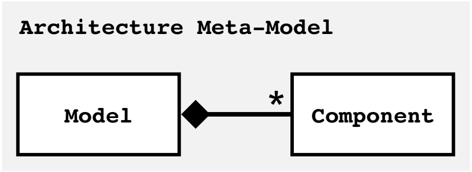
* Classes
* Datatypes
* Attributes
* References
* Operations

References

* Association, e.g,
  * use element
  * extend class
  * apply aspect
  * specialize
  * instantiate
* Containment -> is part of
* Aggregation, e.g.,
  * relates to
  * collects

Approach for Modularization

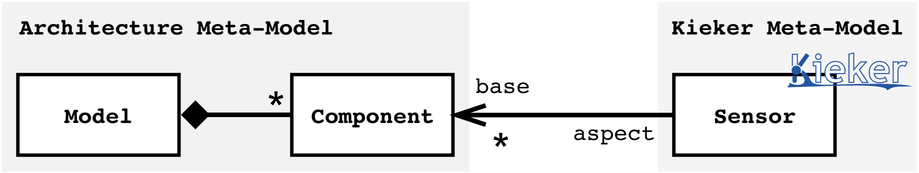
Base and Aspect Meta-Models



VAO 2014 [Jun+14]

Approach for Modularization

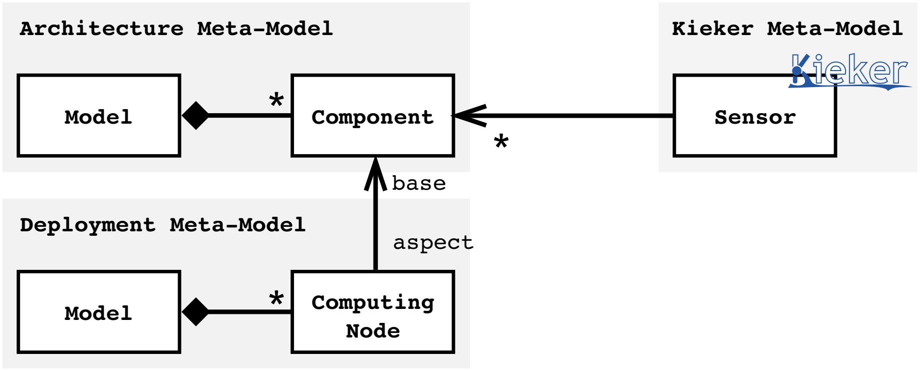
Base and Aspect Meta-Models



VAO 2014 [Jun+14]

Approach for Modularization

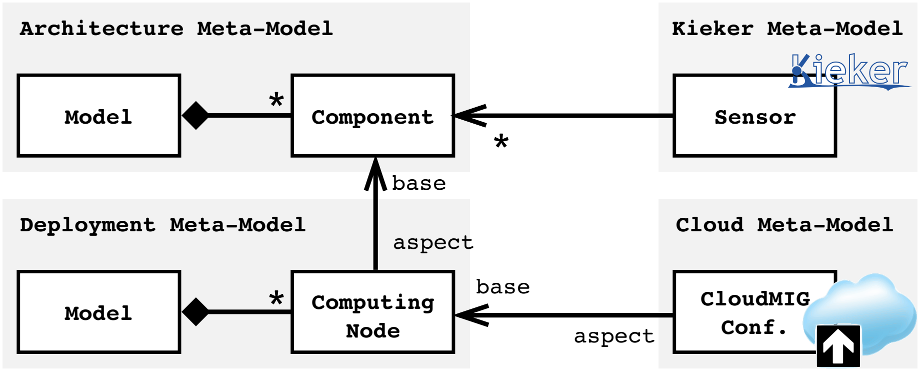
Base and Aspect Meta-Models



VAO 2014 [Jun+14]

Approach for Modularization

Base and Aspect Meta-Models



VAO 2014 [Jun+14]

Use-Cases for Meta-Models

Use-Cases

* Editors
* Transformations
* Simulations
* Evaluations
* Run-time Models

Meta-Model Styles

* Traces
* Navigation
* Queries
* Expressions
* Behavior
* Data
* State
* Types
* Utility

VAO 2014 [Jun+14]

Stakeholders and other Obstacles

Code Base

* Coordination of refactoring and adaptation
* Propagate implementation patterns

Model Migration

* Provide upgrade tooling for models
* Allow to chain upgrade transformations

Outreach to Users and Developers

* Convince to participate in change
* Integrate and align with other development efforts

Evolution Plan

Tasks

- Preparation
- Modernization
- Modularization

Coordination

- Release cycle
- Concurrent development
- Iterative agile realization

[Bro74]

Evolution: Preparation and Modernization

Task A Preparation

* Realize common tool to transform old to new models (*BA*)

Task B Modernization

* Fix issue with **NamedElement.entityName -> NamedElement.name**

Evolution: Preparation and Modernization

Task A Preparation

* Realize common tool to transform old to new models (*BA*)

Task B Modernization

* Fix issue with **NamedElement.entityName -> NamedElement.name**
* Modernize naming scheme of references (only refactoring)

Evolution: Preparation and Modernization

Task A Preparation

* Realize common tool to transform old to new models (*BA*)

Task B Modernization

* Fix issue with **NamedElement.entityName -> NamedElement.name**
* Modernize naming scheme of references (only refactoring)
* Replace explicit container reference by implicit EMF container reference

Evolution: Preparation and Modernization

Task A Preparation

* Realize common tool to transform old to new models (*BA*)

Task B Modernization

* Fix issue with **NamedElement.entityName -> NamedElement.name**
* Modernize naming scheme of references (only refactoring)
* Replace explicit container reference by implicit EMF container reference
* Remove utility references from the meta-model

Evolution: Preparation and Modernization

Task A Preparation

* Realize common tool to transform old to new models (*BA*)

Task B Modernization

* Fix issue with **NamedElement.entityName -> NamedElement.name**
* Modernize naming scheme of references (only refactoring)
* Replace explicit container reference by implicit EMF container reference
* Remove utility references from the meta-model
* Fix naming of classes, e.g., for instance and types

Evolution: Modularization

Task C Planing (iteratively, in parts)

- **Clearly define** concerns and views
    - Views may comprise and aggregate information from different concerns
    - Some concerns have closer bonds than others
- Separate cross-cutting concerns (orthogonal views)

Task D Execution (iteratively, *BA/MA*)

- Check for irregular references between separated aspects/views
- Remove irregular references
- Place separated aspect in a separate meta-model
- Adapt tooling / rewrite editors
- Provide specific transformation tool (for each release)

SWOT Analysis

Challenge Evolution of PCM into an extensible and modular meta-model

Strengths

* Evolution plan
* Modularization approach
* Modularization concept

Weaknesses

* No evolution governance
* Autonomous research groups
* Short term interests

Opportunities

* PCM as platform
* Better incorporation of research results
* Foster adoption in the field through reuse of models and meta-models

Threats

* No developer commitment
* Alienating users

Conclusion

Challenge Evolution of PCM into an extensible and modular meta-model

Solution

* Applying design principles for meta-models
* Implementation plan of the PCM evolution

Future Work

* Organize governance
* Define and execute small evolution steps
* Make outreach to users and developers
* Integrate with release cycle
* **Get started!**

Appendix

Meta-Model Statistics

Summary of pcm.ecore

- Packages 20
- Classes 147
  - abstract 33
  - other 114
  - uncontained 23
- References 284
  - containment 89
  - container 85
  - cyclic 8

Uncontained Concrete Classes

∗ pcm.core.entity.ResourceInterfaceRequiringEntity	
∗ pcm.core.entity.ResourceInterfaceProvidingEntity	
∗ pcm.core.entity.ResourceInterfaceProvidingRequiringEntity	
∗ pcm.usagemodel.UsageModel	**usage model root**
∗ pcm.repository.Repository	**repository root**
∗ pcm.resourcetype.ResourceRepository	
∗ pcm.parameter.CharacterisedVariable	
∗ pcm.seff.CallReturnAction	
∗ pcm.system.System	**system model root**
∗ pcm.resourceenvironment.ResourceEnvironment	
∗ pcm.allocation.Allocation	**allocation model root**

Container Reference Example

pcm.core.PCMRandomVariable

- closedWorkload_PCMRandomVariable : ClosedWorkload
- passiveResource_capacity_PCMRandomVariable : PassiveResource
- variableCharacterisation_Specification : VariableCharacterisation
- infrastructureCall__PCMRandomVariable : InfrastructureCall
- resourceCall__PCMRandomVariable : ResourceCall
- parametricResourceDemand_PCMRandomVariable : ParametricResourceDemand
- loopAction_PCMRandomVariable : LoopAction
- guardedBranchTransition_PCMRandomVariable : GuardedBranchTransition
- specifiedExecutionTime_PCMRandomVariable : SpecifiedExecutionTime
- eventChannelSinkConnector__FilterCondition : EventChannelSinkConnector
- assemblyEventConnector__FilterCondition : AssemblyEventConnector
- loop_LoopIteration : Loop
- openWorkload_PCMRandomVariable : OpenWorkload
- delay_TimeSpecification : Delay
- communicationLinkResourceSpecifcation_throughput_PCMRandomVariable : CommunicationLinkResourceSpecification
- processingResourceSpecification_processingRate_PCMRandomVariable : ProcessingResourceSpecification
- communicationLinkResourceSpecification_latency_PCMRandomVariable : CommunicationLinkResourceSpecification

Reference Naming Schemes

* referenceName__DefiningClassName
* referenceName_DefiningClassName
* referenceName_ArbitraryName
* referenceName__ArbitraryName
* referenceName_referenceName_DefiningClassName

Reference Naming Examples 1/3

pcm.core.entity.InterfaceProvidingEntity

* providedRoles_InterfaceProvidingEntity : ProvidedRole

pcm.core.entity.ResourceProvidedRole

* resourceInterfaceProvidingEntity__ResourceProvidedRole : ResourceInterfaceProvidingEntity (association)
* providedResourceInterface__ResourceProvidedRole : ResourceInterface

pcm.core.entity.ResourceInterfaceProvidingEntity

* resourceProvidedRoles__ResourceInterfaceProvidingEntity : ResourceProvidedRole (containment, finite)

Reference Naming Examples 2/3

pcm.core.PCMRandomVariable

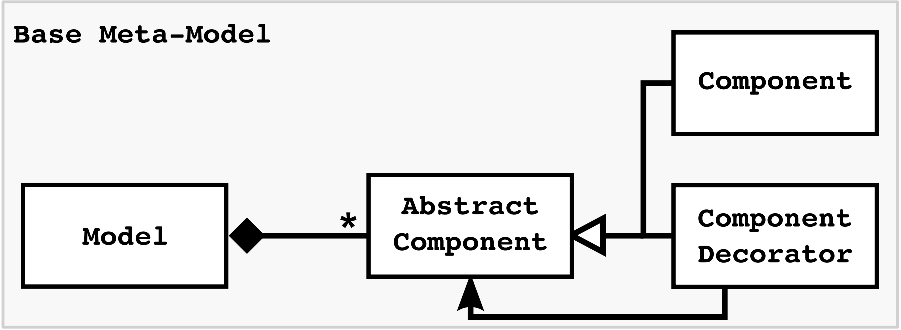
- closedWorkload_PCMRandomVariable : ClosedWorkload
- passiveResource_capacity_PCMRandomVariable : PassiveResource
- variableCharacterisation_Specification : VariableCharacterisation
- infrastructureCall__PCMRandomVariable : InfrastructureCall
- resourceCall__PCMRandomVariable : ResourceCall
- parametricResourceDemand_PCMRandomVariable : ParametricResourceDemand
- loopAction_PCMRandomVariable : LoopAction
- guardedBranchTransition_PCMRandomVariable : GuardedBranchTransition
- specifiedExecutionTime_PCMRandomVariable : SpecifiedExecutionTime
- eventChannelSinkConnector__FilterCondition : EventChannelSinkConnector
- ...

Reference Naming Examples 3/3

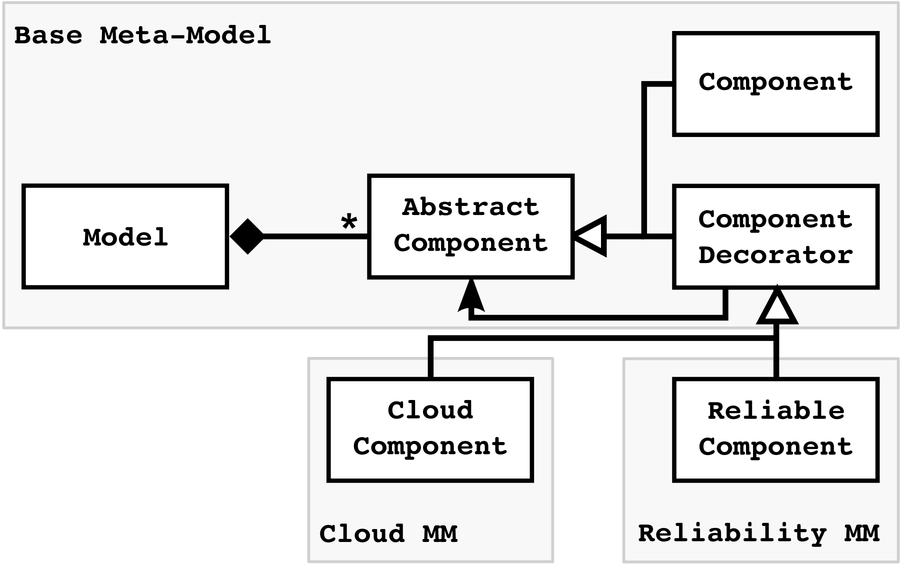
pcm.core.PCMRandomVariable

* assemblyEventConnector__FilterCondition : AssemblyEventConnector
* loop_LoopIteration : Loop
* openWorkload_PCMRandomVariable : OpenWorkload
* delay_TimeSpecification : Delay
* communicationLinkResourceSpecifcation_throughput_PCMRandomVariable : CommunicationLinkResourceSpecification
* processingResourceSpecification_processingRate_PCMRandomVariable : ProcessingResourceSpecification
* communicationLinkResourceSpecification_latency_PCMRandomVariable : CommunicationLinkResourceSpecification

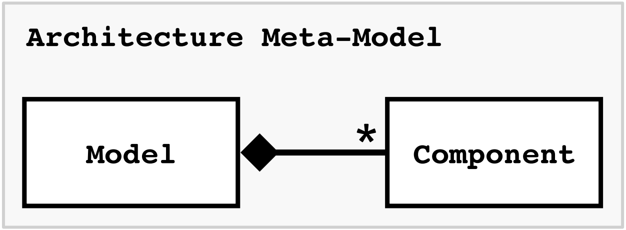
Decorator Pattern



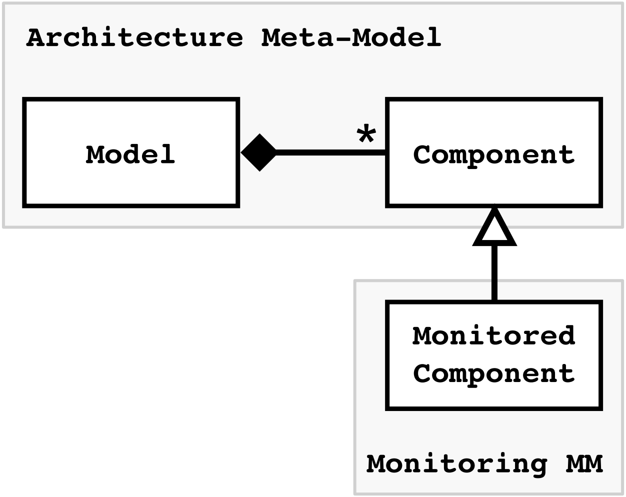
Decorator Pattern



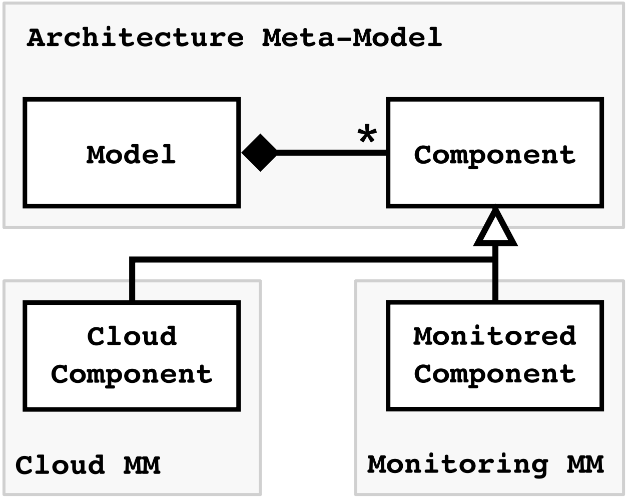
EMF Extension



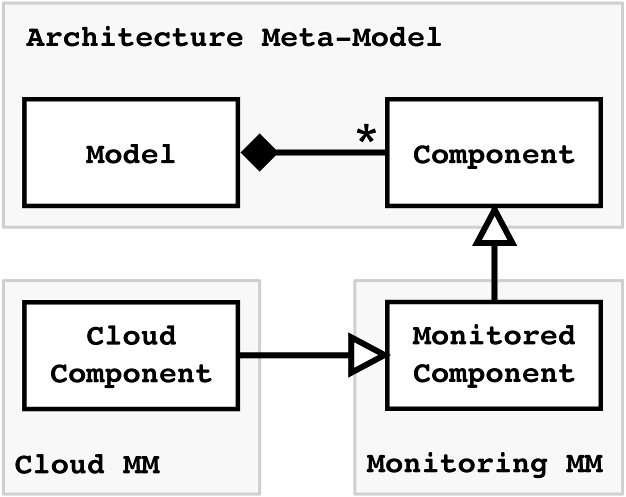
EMF Extension



EMF Extension



EMF Extension



Bibliography I

Becker S., Koziolek H., Reussner R. (2009). “The Palladio Component Model for Model-driven Performance Prediction”. In: J. Syst. Softw. 82.1, 3-22. ISSN: 0164-1212. 10.1016/j.jss.2008.03.066.

Browne D. (1974). “Hägar the Horrible #1”. Tempo.

Gamma E. (1994). “Design patterns: elements of reusable object-oriented software”. Pearson Education.

Jung R. (2014). “A Method for Aspect-oriented Meta-Model Evolution”. In: *Proceedings of the 2Nd Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling*. VAO ’14. York, United Kingdom: ACM, 19:19-19:22. 978-1-4503-2900-2. 10.1145/2631675.2631681. http://doi.acm.org/10.1145/2631675.2631681.

Bibliography II

Kramer M. E. (2012). “Extending the Palladio Component Model using Profiles and Stereotypes”. In: *Palladio Days 2012 Proceedings (appeared as technical report)*. Ed. by Steffen Becker et al. Karlsruhe Reports in Informatics ; 2012,21. Karlsruhe: KIT, Faculty of Informatics, 7-15. http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/2350659.

Langer P. (2012). “EMF Profiles: A Lightweight Extension Approach for EMF Models.” In: Journal of Object Technology 11.1, 1-29. http://dblp.uni-trier.de/db/journals/jot/jot11.html#LangerWWC12.

Steinberg D. (2009). “EMF: Eclipse Modeling Framework”. 2. Boston, MA: Addison-Wesley. 978-0-321-33188-5. http://my.safaribooksonline.com/9780321331885.

Strittmatter M. (2013). “Towards a Modular Palladio Component Model”. In: *KPDAYS*, 49-58.