Evolution of the Palladio Component Model: Process and Modeling Methods

SOSP 2014 - Stuttgart

Reiner Jung\textsuperscript{1} Philipp Merkle\textsuperscript{2} Misha Strittmatter\textsuperscript{2}

\textsuperscript{1} Kiel University
\textsuperscript{2} Karlsruhe Institute of Technology

28\textsuperscript{th} November 2014

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

**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
Solution

Approach for Modularization

Base and Aspect Meta-Models

Architecture Meta-Model

Model * Component
Approach for Modularization

Base and Aspect Meta-Models

Architecture Meta-Model

Model * Component

Kieker Meta-Model

Sensor

base aspect

VAO 2014 [Jun+14]
Approach for Modularization

Base and Aspect Meta-Models

Architecture Meta-Model
- Model
- Component
  * From Model to Component

Kieker Meta-Model
- Sensor
  * From Component to Sensor

Deployment Meta-Model
- Model
- Computing Node
  * From Model to Computing Node
  base
  aspect

VAO 2014 [Jun+14]

Jung, Heinrich, Merkle, Strittmatter
Approach for Modularization

Base and Aspect Meta-Models

Architecture Meta-Model

- Model
  - * Component

Kieker Meta-Model

- Sensor
  - *

Deployment Meta-Model

- Model
  - * Computing Node
    - base
    - aspect

Cloud Meta-Model

- CloudMIG Conf.
  - base
  - aspect

VAO 2014 [Jun+14]
Solution

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
> Solution

**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]
Task A Preparation

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

Task B Modernization

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

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)
Solution

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
Solution

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)
## Conclusion

### SWOT Analysis

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

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Evolution plan</td>
<td>* No evolution governance</td>
</tr>
<tr>
<td>* Modularization approach</td>
<td>* Autonomous research groups</td>
</tr>
<tr>
<td>* Modularization concept</td>
<td>* Short term interests</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>* PCM as platform</td>
<td>* No developer commitment</td>
</tr>
<tr>
<td>* Better incorporation of research results</td>
<td>* Alienating users</td>
</tr>
<tr>
<td>* Foster adoption in the field through reuse of models and meta-models</td>
<td></td>
</tr>
</tbody>
</table>
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
* pcm.repository.Repository
* pcm.resourcetype.ResourceRepository
* pcm.parameter.CharacterisedVariable
* pcm.seff.CallReturnAction
* pcm.system.System
* pcm.resourceenvironment.ResourceEnvironment
* pcm.allocation.Allocation
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
* communicationLinkResourceSpecification_throughput_PCMRandomVariable : CommunicationLinkResourceSpecification
* processingResourceSpecification_processingRate_PCMRandomVariable : ProcessingResourceSpecification
* communicationLinkResourceSpecification_latency_PCMRandomVariable : CommunicationLinkResourceSpecification
Appendix

Reference Naming Schemes

* referenceName__DefiningClassName
* referenceName_DefiningClassName
* referenceName_ArbitraryName
* referenceName__ArbitraryName
* referenceName_referenceName_DefiningClassName
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)
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
* ...


**pcm.core.PCMRandomVariable**

* assemblyEventConnector__FilterCondition : AssemblyEventConnector

* loop_LoopIteration : Loop

* openWorkload_PCMRandomVariable : OpenWorkload

* delay_TimeSpecification : Delay

* communicationLinkResourceSpecification_throughput_PCMRandomVariable : CommunicationLinkResourceSpecification

* processingResourceSpecification_processingRate_PCMRandomVariable : ProcessingResourceSpecification

* communicationLinkResourceSpecification_latency_PCMRandomVariable : CommunicationLinkResourceSpecification
Appendix

Decorator Pattern

Base Meta-Model

Model

Abstract Component

Component

Component Decorator

Jung, Heinrich, Merkle, Strittmatter
Appendix

Decorator Pattern

Base Meta-Model

Model

Abstract Component

Component

Component Decorator

Cloud Component

Reliable Component

Cloud MM

Reliability MM

Jung, Heinrich, Merkle, Strittmatter
EMF Extension

Architecture Meta-Model

Model

Component

*
Appendix

EMF Extension

Architecture Meta-Model

Model \* Component

Monitored Component

Monitoring MM
EMF Extension

Architecture Meta-Model

Model -> Component

Cloud Component
Cloud MM

Component

Monitored Component
Monitoring MM

Jung, Heinrich, Merkle, Strittmatter
EMF Extension

Architecture Meta-Model

Model

Component

* 

Cloud Component

Cloud MM

Monitored Component

Monitoring MM

Jung, Heinrich, Merkle, Strittmatter


