Extending the Palladio Component Model using Profiles and Stereotypes

Palladio Days 2012, Paderborn

Motivation - Organic Growth of the PCM

SimuLizar  ProbeSpec
Completions   EventSim
Design Patterns & Rationale  SimuCom
KAMP
PerOpteryx
PCM
Data Flow Security
SLAstic.SIM

Motivation  Concept  Implementation  Example Extensions  Conclusion

Motivation - Organic Growth of the PCM

SimuLizar, ProbeSpec, Completions, Design Patterns & Rationale, KAMP, EventSim, SimuCom, Data Flow Security, SLAStic.SIM

Motivation
Concept
Implementation
Example Extensions
Conclusion


November 8, 2012
Motivation - Organic Growth of the PCM

SimuLizar  ProbeSpec
Complections  EventSim
Design Patterns & Rationale
KAMP
PerOpteryx
PCM
Data Flow Security
SLAStic.SIM
Example extension: design rationale

```
:AllocateSeparately
Decision

allocationContexts[0]

A:ResourceContainer
:ComponentA

allocationContexts[1]

B:ResourceContainer
:ComponentB

invariant:
self.allocationContexts
  ->forall(ac1, ac2| ac1 <> ac2 implies
  ac1.resourceContainer <>
  ac2.resourceContainer
```
Motivation - Requirements

Requirements for a PCM extension mechanism

- **type-safe** typed extensions according to type rules
- **non-invasive** no adaptation of the PCM for further extensions
- **lightweight** extending has to be easier than complete metamodelling
- **flexible** straightforward extensions as well as complex extensions
- **intuitive** based on notations that are familiar to many developers
- **composable** new extensions can rely on existing ones
Extending PCM using profiles & stereotypes

- Similar to UML: profiles consist of stereotypes
- Common syntax for adding information to PCM elements
- Enables generic extension tooling within Palladio
- Can be realized using EMF Profiles [LWWC11, LWWC12]
Example extension profile

```
<<Stereotype>>
AllocateSeparatelyDecision

<<extends>>
allocationContexts

<<condition>>
self.allocationContexts -> forAll(
    ac1, ac2 | ac1 <> ac2 implies ac1.resourceContainer <> ac2.resourceContainer
)
```

Motivation  Concept  Implementation  Example Extensions  Conclusion
Concept - Overview

PCM Profiles - Models, layers and relations

- Ecore
- Profiles MM
- aProfile
- aProfileApplication

Metamodel Level:
- PCM
- Profiles MM
- aProfile

Extends:
- aBaseModel
- aProfileApplication

Motivation Concept Implementation Example Extensions Conclusion


November 8, 2012
PCM Profiles - Models, layers and relations

- Ecore
- Profiles MM
- aProfile
- aProfileApplication
- «instanceOf»
- «instanceOf»
- «instanceOf»
- M1
- M2
- M3

Metamodel Level

- PCM
- Profiles MM
- Domain MM

- aBaseModel
- aProfile
- «instanceOf»
- «instanceOf»
- extends
- references

Motivation Concept Implementation Example Extensions Conclusion


November 8, 2012
Two step extension of the PCM

- Specifying a profile in the editor
  - Extension relations & OCL constraints
  - Attributes & references
  - Dependencies between stereotypes
- Registering the profile

Generic use of PCM extensions

- Automatic integration into tree-based & graphical editors
- Stereotypes applicable if restrictions are satisfied
- Default view for editing stereotype applications
- Separate storage of profile & stereotype applications
Generic extension of the graphical editor

Motivation
Concept
Implementation
Example Extensions
Conclusion


November 8, 2012
Example Extension - Dataflow Security

Without an extension mechanism

Motivation Concept Implementation Example Extensions Conclusion


November 8, 2012
Using PCM Profiles

<<Stereotype>>

SecureContainer

<<Stereotype>>

Attacker

pcm

1..1

ResourceContainer

0..*

UsageScenario

1..1
### Requirements fulfilled by our approach

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type-safe</td>
<td>typed extends relations and OCL constraints</td>
</tr>
<tr>
<td>non-invasive</td>
<td>profile definition and application without changing the PCM</td>
</tr>
<tr>
<td>lightweight</td>
<td>specific concepts and tooling for stereotype applications</td>
</tr>
<tr>
<td>flexible</td>
<td>annotation-like profiles with simple-typed information as well as complex profiles referring to extension metamodels</td>
</tr>
<tr>
<td>intuitive</td>
<td>editor support &amp; notation similar to UML profiles</td>
</tr>
<tr>
<td>composable</td>
<td>profiles and stereotypes can depend on existing ones</td>
</tr>
</tbody>
</table>
PCM Profiles - PIBA

Problem
Extending PCM = repeating information & tooling in an unstructured way

Idea
Propose a uniform extension mechanism and provide infrastructure

Benefit
Reduces effort for future extensions & makes them more reusable

Action
Identify requirements and propose a profile- & stereotype-based approach
References


Without an extension mechanism
Without an extension mechanism

\[
\begin{array}{c}
\text{ProbeSet} \\
\text{position: ProbeSetPosition} \\
1 \text{..* probes} \\
\text{Probe} \\
\end{array}
\begin{array}{c}
\text{annotatedElement} \\
\text{<<enumeration>>} \\
\text{ProbeSetPosition} \\
\text{BEFORE} \\
\text{AFTER} \\
\text{NONE} \\
\end{array}
\begin{array}{c}
\text{EObject} \\
1 \\
\end{array}
\]
Backup - UML & EMF Profiles

UML Profiles
- Profile & stereotype application stored within extended model
- No dependencies between stereotypes

EMF Profiles
- Persistence in publicly visible XMI-files
- No dependencies between stereotypes
- No OCL constraints restricting stereotype application
- No packages within profiles
- No inheritance between stereotypes
- No evaluation of multiplicity constraints
EMF Profiles - Models, layers and relations

- **M3**: Ecore
  - «instanceOf»

- **M2**: EMF Profiles MM
  - «inheritsFrom»
  - «instanceOf»

- **M1**: aProfile
  - «instanceOf»
  - «instanceOf»
  - «instanceOf»

**Metamodel Level**

- Metalevel lifting by inheritance

---

*M. E. Kramer, Z. Durdik, M. Hauck, J. Henss, M. Küster, P. Merkle and A. Rentschler – Extending the Palladio Component Model using Profiles and Stereotypes*