Non-Invasive Palladio Development

Palladio Days 2011

Michael Hauck  hauck@fzi.de
Benjamin Klatt  klatt@fzi.de
Motivation

- The Palladio Tooling codebase becomes larger and larger
- More people and more groups are using and developing Palladio
- Different steps to ensure a maintainable codebase have to be taken (or have already been taken)
- In this talk, we want to present some extension concepts we introduced into Palladio

Take a simple SimuCom run – this is what we want:

- Load PCM Models
- Load Middleware Configuration
- Validate PCM Models
- Transform for Accuracy Analysis
- Transform for Sensitivity Analysis
- Apply Completions
- Transform Events
- Initialize Reliability
- Apply hard-coded simulation completion (network connectors)
- Generate and compile SimuCom code
- Execute SimuCom code
Extending the SimuCom Workflow

- With an extensible workflow, workflow jobs can be defined in a „non-invasive“ way
  - PCM Core functionality does not depend on extensions
    - Smaller core build
  - Extensions are optional (only install and activate if needed)
  - Extension also help to structure code belonging together
What We Did Last Summer

A new code repository structure

The traditional PCM plugins have been moved to Core

New PCM enhancements should be developed in the Incubation area to avoid breaking the PCM build

Addons area: For projects which extend PCM, but on which the PCM core functionality does not depend
Extending a Workflow:

Job A
- Defines workflow extension ID
- Extensions can register by using Eclipse extension point mechanism

Extendable Job B
- Extendable Job calls registered extensions

Job C
- Implements extension point: Provides
  - Workflow ID
  - Configuration tab
  - Configuration object
  - Workflow Engine job
  - Priority

Extending Job D
Extending a SimuCom Workflow:

- Currently, SimuCom provides 3 workflow extension IDs to provide Job extensions:
  - `WORKFLOW_ID_AFTER_LOAD_VALIDATE`
    - Executes jobs directly after initial model has been loaded and validated
  - `WORKFLOW_ID_BEFORE_DOCK`
    - Executes jobs after SimuCom code has been created
  - `WORKFLOW_ID_AFTER_DOCK`
    - Executes jobs just before SimuCom code is about to be executed
Example: Integrating Infrastructure Performance Overheads into SimuCom

- Ginpex tool provides overhead model & SimuCom extension
- If activated, SimuCom extension adds overhead to resource demands

```
ResourceContainer
Issue X CPU demand
Calculate overhead
```

SimuCom CPU Resource
Example: Integrating Infrastructure Performance Overheads into SimuCom

- Influence on prediction accuracy

1 user

50 parallel users

![Graphs showing influence on prediction accuracy for 1 user and 50 parallel users.](image-url)
Open Issues

PCM Model

PCM Model Extension (e.g. decorator model)

extends

M2T transformation (oAW XPAND)

SimuCom Code

SimuCom Extension

extends

extends
Conclusions

- New code repository structure

- Workflow Engine now features extendable jobs – start using them!

- Workflow Engine documentation: http://sdqweb.ipd.kit.edu/wiki/Palladio_Workflow_Engine

- How can we use the new feature (or improve it) to improve the PCM workflow code?