CI/CD with OpenModelica for Library and Tool developers

Andreas Heuermann

Hochschule Bielefeld - University of Applied Sciences and Arts
Institute for Data Science Solutions

5th February 2024
AGENDA

1. CI/CD in OpenModelica eco-system
2. GitHub Workflows
   - setup-openmodelica action
   - openmodelica-library-testing action
3. VS Code Extension
   - Modelica language server
CI/CD IN A NUTSHELL

- Continuous Integration & Continuous Delivery (or Development)
  - CI: Frequent merging of small changes
  - CD: Release software at any time

Basic Idea

- Test before you break stuff!
- Fast release cycles
CI/CD AND OPENMODELICA

Git + GitHub + Jenkins + Docker

- Testing Pull Requests
  - MSYS2 change MINGW64 to UCRT64 #10939
  - Unit test on Windows / Linux / Mac with different toolchains.
  - Ensure I wrack less havoc.
OPENMODELICA LIBRARY COVERAGE TESTS

Modelica_4.0.0 test using OpenModelica

| Total Frontend Backend SimCode Templates Compilation Simulation Verification |
|------------------|------------------|------------------|------------------|------------------|------------------|
| 514              | 514              | 514              | 514              | 514              | 513              | 504              | 486              |

Test started: 2024-01-24 00:52:58
Total time taken: 1:07:43
System info: AMD Ryzen 9 5950X 16-Core Processor, 63 GB RAM, Ubuntu 22.04.3 LTS
OpenModelica Version: OMCompiler v1.23.0-dev.241+g00dc99398b
OpenModelicaLibraryTesting Changes

Commit     Date        Author          Summary
e188a42 2024-01-22 14:42:30 +0100 Andreas Adding CI (#53)
OTHER OPENMODELLICA TOOLS

- Testing OpenModelica APIs
  - OMPython: Python scripting interface
  - OMJulia.jl: Julia scripting interface

- Testing compatibility with 3rd party tools
  - FMI export and import
AGENDA

1. CI/CD in OpenModelica eco-system
2. GitHub Workflows
   - setup-openmodelica action
   - openmodelica-library-testing action
3. VS Code Extension
   - Modelica language server
A workflow is a configurable automated process that will run one or more jobs. Workflows are defined by a YAML file checked in to your repository and will run when triggered by an event in your repository, or they can be triggered manually, or at a defined schedule.

From: https://docs.github.com/en/actions/using-workflows/about-workflows#about-workflows
GITHUB WORKFLOWS AND ACTIONS

- Workflows: Easy-to-integrate CI/CD pipelines on GitHub
- Actions: Reusable packages to perform specific tasks in your workflows.

```
name: Test
on: [push]
jobs:
  unit-test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: OpenModelica/setup-openmodelica@v1
      - run: omc --version

.github/workflows/test.yml
```
AGENDA

1. CI/CD in OpenModelica eco-system
2. GitHub Actions
   - setup-openmodelica action
   - openmodelica-library-testing action
3. VS Code Extension
   - Modelica language server
SETTING UP OPENMODELICA

- Get OpenModelica into your environment

<table>
<thead>
<tr>
<th>Scripts</th>
<th>Docker</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS dependent</td>
<td>Fast</td>
<td>Combine with scripts or</td>
</tr>
<tr>
<td>Easy to set up</td>
<td>Linux only</td>
<td>Docker images</td>
</tr>
<tr>
<td>Difficult to</td>
<td>Keep up-to-date</td>
<td>Reusable</td>
</tr>
<tr>
<td>maintain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SETUP-OPENMODELICA

GitHub Action: OpenModelica/setup-openmodelica

- uses: OpenModelica/setup-openmodelica@v1
  with:
    version: '1.22.1'
    packages: |
      'omc'
      'omsimulator'
    libraries: |
      'Modelica 4.0.0'
      'Modelica 3.2.3+maint.om'
  omc-diff: true
CURRENTLY SUPPORTED

- Linux
  - Advanced Packaging Tool (APT)
  - Most non-GUI OM packages
    - OpenModelica Compiler, OMSimulator, omc-diff, ...
- Windows
  - Full installer
  - Most OpenModelica versions
PLANNED

- Mac support
- Optional Docker container
- Archive libraries
AGENDA

1. CI/CD in OpenModelica eco-system
2. GitHub Actions
   - setup-openmodelica action
   - openmodelica-library-testing action
3. VS Code Extension
   - Modelica language server
OPENMODELICA-LIBRARY-TESTING

GitHub Action: OpenModelica/openmodelica-library-testing-action

- uses: OpenModelica/openmodelica-library-testing@v0.1
  with:
    library: 'MyLibrary'
    library-version: '2.2.0'
    modelica-file: 'MyLibrary/package.mo'
    omc-version: 'stable'
    reference-files-dir: 'ReferenceFiles'
    reference-files-extension: 'mat'
    reference-files-delimiter: '.'
OPENMODELICA-LIBRARY-TESTING

- Easy way to use OpenModelicaLibraryTesting
  - https://github.com/OpenModelica/OpenModelicaLibraryTesting

- Automated conf.json generation
- Setup and run Python scripts
  - Test all models with experiment annotation
  - Compare simulation results to reference
  - Generate HTML coverage report
- Collect and archive results
## PNLIB DEMO

### test (stable) summary

<table>
<thead>
<tr>
<th>Total</th>
<th>Frontend</th>
<th>Backend</th>
<th>SimCode</th>
<th>Templates</th>
<th>Compilation</th>
<th>Simulation</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>92</td>
</tr>
</tbody>
</table>

### Summary

### Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Verified</th>
<th>Simulate</th>
<th>Total buildModel</th>
<th>Parsing</th>
<th>Frontend</th>
<th>Backend</th>
<th>SimCode</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNLib.Examples.ConTest.Conflict (sim)</td>
<td>0.01 (13 verified)</td>
<td>0.05</td>
<td>3.90</td>
<td>2.02</td>
<td>0.11</td>
<td>0.40</td>
<td>0.03</td>
<td>0.09</td>
</tr>
<tr>
<td>PNLib.Examples.ConTest.ConflictLoop (sim)</td>
<td>0.01 (13 verified)</td>
<td>0.35</td>
<td>4.53</td>
<td>2.00</td>
<td>0.11</td>
<td>0.61</td>
<td>0.05</td>
<td>0.12</td>
</tr>
</tbody>
</table>
AGENDA

1. CI/CD in OpenModelica eco-system
2. GitHub Actions
   - setup-openmodelica action
   - openmodelica-library-testing action
3. VS Code Extension
   - Modelica language server
MODELICA LANGUAGE SERVER

Based on
- Language Server Protocol (LSP)
- Tree-sitter parser
  - OpenModelica/tree-sitter-modelica
LANGUAGE SERVER PROTOCOL

Open Modelica file

Modelica Editor

Notification: textDocumentation/didOpen

Notification: textDocumentation/publishDiagnostics

“Goto definition”

Request: textDocumentation/definition

Response: textDocumentation/definition; Result: Location

Modelica LS

Publish errors and warnings
MODELICA LANGUAGE SERVER

- Currently available features
  - Document Symbol Provider

- Planned features
  - [WIP] Documentation on hover
  - Go to Definition
  - Auto complete
  - Sematic Highlighting
CONCLUSION

- GitHub Actions
  - OpenModelica/setup-openmodelica
  - OpenModelica/openmodelica-library-testing-action

- VS Code Extension
  - OpeModelica/modelica-language-server
  - Soon™: OpeModelica/metamodelica-language-server

Contributions are welcome
The presented work is part of the PHyMoS project, supported by the German Federal Ministry for Economic Affairs and Climate Action.

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag

Project number: 19I20022G

Proper Hybrid Models for Smarter Vehicles