EclipseCon Europe 2015

An Open Source Development Platform
For Embedded Multi- and Many-Core Systems

Harald Mackamul, Robert Bosch GmbH
APP4MC – Application Platform Project for MultiCore

- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps
APP4MC – Application Platform Project for MultiCore

- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps
AMALTHEA Timeline

ITEA2 Project
itea3.org/project/amalthea.html

Developer Preview
Sept 2013

Release 1.0.3
June 2014

Release 1.1.0
Aug 2015

Release 1.1.1
Oct 2015

ITEA2 Project
itea3.org/project/amalthea4public.html

Eclipse Project
Community

All releases were published under the Eclipse Public License (EPL)

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
AMALTHEA
Latest Open Source Releases

AMALTHEA
An Open Platform Project for Embedded Multicore Systems

Release 1.1.1
October 2015

http://www.amalthea-project.org

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
APP4MC – Application Platform Project for MultiCore

- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps
Challenges of Embedded Multi-Core

Software

Hardware

Software Distribution

Memory Mapping

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Challenges of Embedded Multi-Core

- **Typical Paradigm of Single-Core Software**
  - Blackboard Architecture: Memory access is for “free”
  - Integration challenge: scheduling of computation

- **Paradigm Change for Multi-/Many-Core**
  - Cross-Core Communication is expensive
  - Synchronization leads to high overheads
  - Memory location matters
  - Integration challenge: scheduling of computation and communication

- Sophisticated new tooling required for task distribution, memory location optimization and performance analysis

Cross-core communication is a new resource bottleneck
### From Multi- to Many-Core

#### Multi-Core
- Small number of homogenous cores with shared memory
- Mostly symmetric connectivity (e.g. crossbar)
- Limited impact on SW distribution

#### Many-Core
- Larger number of heterogeneous cores with distributed memories
- Increasingly heterogeneous connectivity (Non-uniform Memory Access)
- High impact on SW distribution

Today's automotive Multi-Cores already have Many-Core characteristics
Agenda

APP4MC – Application Platform Project for MultiCore

- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems

- The AMALTHEA Platform
  - High level description
  - Use cases
  - Connections to other Eclipse projects
  - Technical decisions and experiences

- Demo / Screenshots of current release
- APP4MC - Next steps
Constraints
Period $T_1 = 2\text{ms}$
Deadline $D_1 = 1.5\text{ms}$
Period $T_2 = 5\text{ms}$
Deadline $D_2 = 5\text{ms}$

Costs
$T_1$ takes $10\mu\text{s}$ on Core0, $20\mu\text{s}$ on Core3

Decisions
Run $T_1$ on Core0
Run $T_2$ on Core1
Offset of $T_2 = 1\text{ms}$

HW Platform

SW Application

AMALTHEA4public

AMALTHEA
System Model

Simulation

“Performance Simulation” *

* Focus on Timing, Scheduling

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Example of a timing / scheduling simulation*

*Commercial tool – not part of the open source project
Tool platform AMALTHEA
Processing, Simulation and Analysis

SW Modeling
Initial model of software behavior

AMALTHEA System Model

Partitioning
Identification of initial tasks

System Modeling
- Hardware
- Constraints

Simulation
Software Execution

Optimization
- Task distribution
- Memory mapping

AMALTHEA Traces
Tool platform AMALTHEA
Use cases @ BOSCH

Software Structure
Runtime Information
HW Architecture

extract

AMALTHEA
System Model

BOSCH
Simulation & Analysis

exchange

AMALTHEA
System Model

software data

AMALTHEA
System Model

Simulator & Analysis
Tool platform AMALTHEA
Platform Architecture

3rd Party Tools / Custom Tools

AMALTHEA Tools
- Variant Tool
- V & V
- Partitioning & Mapping
- Component Editor

AMALTHEA Tool Platform

Joint Model Repository

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Tool platform AMALTHEA
Connection to other Eclipse Projects

Diagram showing connections between SW Modeling, System Modeling, Partitioning, Simulation Execution, Optimization, Traces, and other projects such as mbeddr, Damos, TRACE COMPASS, Franca, EATOP, and RMF.
Tool platform AMALTHEA
Connection to other Eclipse Projects

planned extension
- Timing Annotations
Tool platform AMALTHEA
Connection to other Eclipse Projects

Franca

to describe

- Interfaces

```plaintext
interface BlowerCtrl_power {
  attribute UInt16 blwOutValuereadonly noSubscriptions
}

interface BlowerCtrl_ctl {
  attribute UInt16 passThroughValue
  broadcast autoMode {}
  broadcast passThroughMode {}
}

interface Temperature {
  attribute UInt16 temp readonly noSubscriptions
}

interface TemperatureAdapter_Signals {
  broadcast turnOn {}
  broadcast turnOff {}
}

interface TemperatureAdapter_Ticks extends HMI_commToggle {
  attribute UInt8 plusTicksreadonly noSubscriptions
  attribute UInt8 minusTicksreadonly noSubscriptions
}
```
Tool platform AMALTHEA
Connection to other Eclipse Projects

implemented

- BTF Import
- BTF View

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Tool platform AMALTHEA
Eclipse Technology

- Eclipse IDE for Automotive Software Developers
- Xcore
  -> Definition of the AMALTHEA Ecore model
- Sphinx
  -> Workspace Handling, Validation, Tree Editor
- Xtend2
  -> Code generation, model transformation
Tool platform AMALTHEA
Eclipse Technology

- **Lyo**
  
  -> *OSLC* (separate Java libraries, no bundles, not included in orbit)

- **Ecore Tools**

  -> *Class diagrams* (only limited connection to Xcore models)

- **Sirius**

  -> *Graphical editors (first prototype)*
APP4MC – Application Platform Project for MultiCore

- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps
Platform Demo
AMALTHEA Model Editor

© APP4MC - All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Platform Demo
Eclipse Help – Examples

The AMALTHEA tool platform comes with several examples. This section will describe how a new project based on these examples can be created.

Step 1

The first step is to create a project. To do this, click the “new” icon in the top left corner and select “Example...”.

Step 2
Mapping example 1

The following section will describe how to calculate a mapping with the AMALTHEA tool platform using the DFG and ILP based load balancing algorithms.

Step 1
Software Model

The AMALTHEA software model is central accessible through the SW/Model element. The namespace for the model is "http://www.amalthea.itea2.org/model/1.3.0/sw".

Memory Information

Analyzing and mapping the software structure to available memories needs additional information of the included elements. This type of information targets the consumed size of memory of an element, represented by the size attribute of type DataUnit. The element AbstractElementMemoryInformation is a generalized element that provides this data. The following image shows the structure and also the elements of the software model that are extending AbstractElementMemoryInformation (the overview picture is only showing the hierarchy and not possible relationships between the elements):
Platform Demo
Eclipse Help – Developer Guide

For all of these features there also exists an SDK containing the sources. If you install the
Adding a new workflow component

Below you will find a sample how to add and implement a new workflow component.

Create project

1. Add a new plugin project with the name `my_sample.workflow`
2. Open the `MANIFEST.MF` in the META-INF folder.
3. Switch to the tab Dependencies to add the following plugin dependencies:
   - `org.eclipse.core.runtime`
   - `org.itea2.amalthea.model.central`
   - `org.itea2.amalthea.model.common`
   - `org.itea2.amalthea.model.components`
   - `org.itea2.amalthea.model.config`
   - `org.itea2.amalthea.model.constraints`
   - `org.itea2.amalthea.model.events`
   - `org.itea2.amalthea.model.hw`
   - `org.itea2.amalthea.model.mapping`
   - `org.itea2.amalthea.model.os`
   - `org.itea2.amalthea.model.propertyconstraints`
   - `org.itea2.amalthea.model.stimuli`
   - `org.itea2.amalthea.model.sw`
   - `org.itea2.amalthea.workflow.base`
   - `org.eclipse.omf.mwo2.launch`
   - `org.eclipse.omf.mwo2.lib`
4. Add a new class `my_sample.workflow.HelloWorld`, which is extending
   - `org.itea2.amalthea.workflow.base.AmaltheaWorkflow`
5. Implement something in the `invokeInternal` method (see sample below).

```java
@Override
protected void invokeInternal(final WorkflowContext ctx, final ProgressMonitor monitor, final String workflowId, String[] ids, long[] time, String[] progress, String[] error)
// some checking if sw model is available
if (null == getAmaltheaModel(ctx).getSwModel()) {
    issues.addError(this, "No proper SWModel available!", getModelSlot());

```
AMALTHEA Model XSD Schema

The AMALTHEA model is specified with the Eclipse Modeling Framework (EMF). In EMF data-models are defined with the Ecore/Xcore format. Eclipse developers can use the AMALTHEA Platform to work with the model. For developers that are not using Java or Eclipse, the specification is also provided in the XSD format. A XSD file for each package can be found here.

The AMALTHEA model references two external model definitions:
- The Ecore model (http://www.eclipse.org/emf/2002/Ecore)
- The Franca model (http://core.franca.org)

The XSD file for the Ecore model can be found on the Eclipse EMF download page.
Download the Models package from Latest Releases. The download contains a folder “org.eclipse.emf.ecore_” This folder contains the XSD schema for Ecore.

The Franca model is only available as Ecore file. To generate an XSD schema open the AMALTHEA Tool Platform and do the following steps:

Open the Plug-Ins View by selecting Window -> Show View -> Other and select Plug-Ins View in the folder Plug-in Development.
Querying for the service provider catalog

In OSLC terminology the service provider catalog is the main entry point of the tool adapter. The service provider catalog provides URLs for querying all the Amalthea models that are available inside the mode directory.

The query URI is of the following form:

http://hostname:port/basePath/catalog

Where basepath is the query path chosen by the user. By default its value is: amaltheaadapter/services.

For example:

http://localhost:8181/amaltheaadapter/services/catalog

The following screenshots show how the firefox Rest Client is used to query the OSLC adapter for the Service provider catalog.
Agenda

APP4MC – Application Platform Project for MultiCore

- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps
Project Activities
Eclipse Project APP4MC

http://projects.eclipse.org/projects/technology.app4mc
Project Activities
Eclipse Project APP4MC

- Open Source **Eclipse project** is created
- Current work: Committer Agreements, Code refactoring, IP clearance, ...
Project Activities
Timeline & future activities

- Architecture optimization
- Conceptual / Implementation
- Official Eclipse project
- Implementation
- Demonstrator
- Community
- Enhancement
- Maintenance
- Optimization

2015 2016 2017
Evaluate the session

Evaluate the sessions at www.eclipsecon.org

+1  0  -1