



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

Harald Mackamul, Robert Bosch GmbH

SPONSORED BY THE



Federal Ministry
of Education
and Research

APP4MC – Application Platform Project for MultiCore

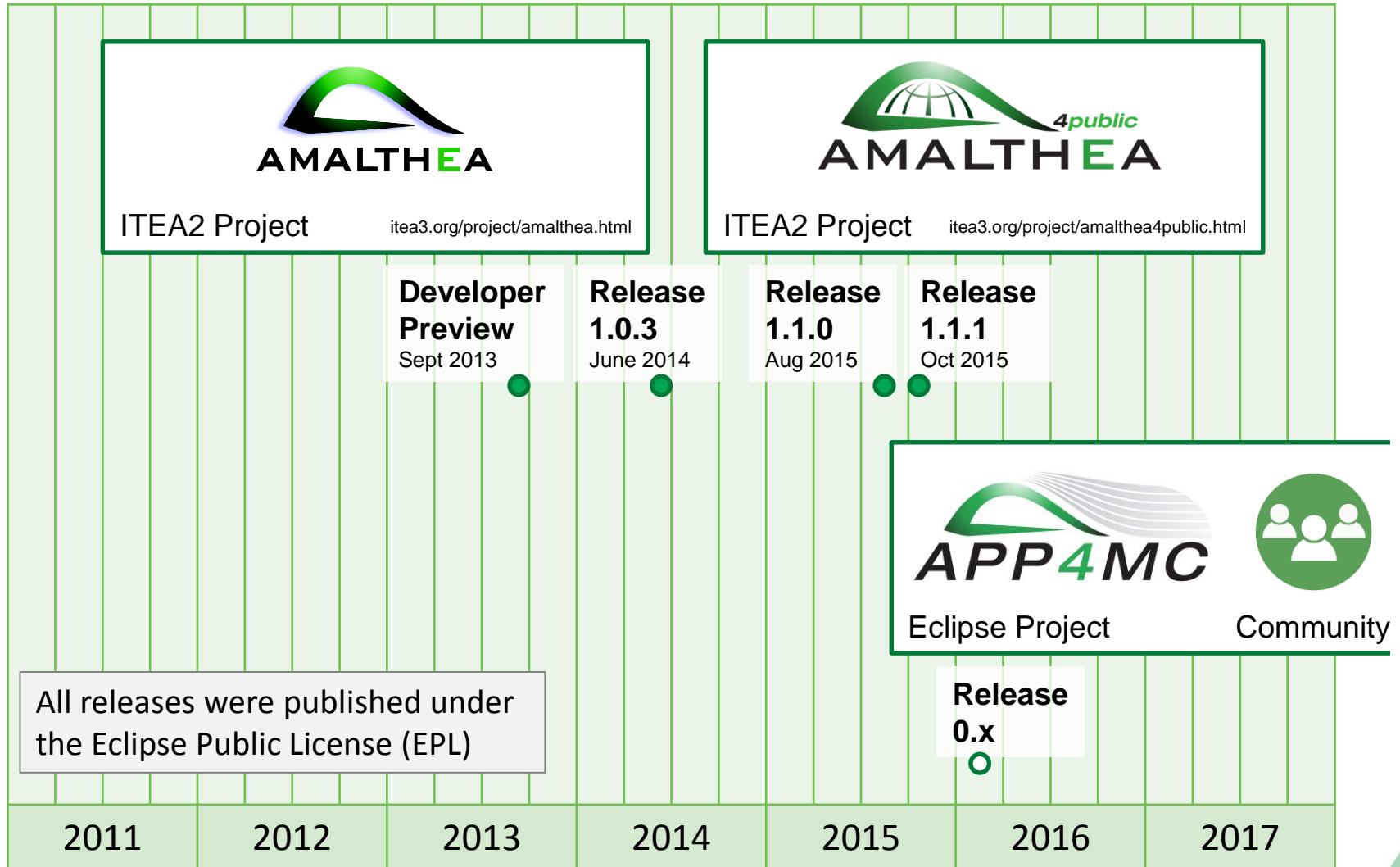
- AMALTHEA - Timeline and current project(s)
- Challenges for embedded multi- and many-core systems
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps

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




AMALTHEA

Latest Open Source Releases





AMALTHEA
An Open Platform Project for Embedded Multicore Systems



- Home
- Downloads
- Results
- News
- Roadmap
- Contact
- Forum

Downloads

Name	Version	Size	Hits
amalthea-win32.win32.x86_64-1.1.1.201510151048	Version 1.1.1 for Windows (64 Bit)	301.42 MB	15
amalthea-win32.win32.x86-1.1.1.201510151048	Version 1.1.1 for Windows (32 Bit)	301.28 MB	0
amalthea-linux.gtk.x86_64-1.1.1.201510151048	Version 1.1.1 for Linux (64 Bit)	300.88 MB	9
amalthea-macosx.cocoa.x86_64-1.1.1.201510151048	Version 1.1.1 for OSX (64 Bit)	300.58 MB	1

Archived Downloads

Name	Version	Size	Hits
amalthea-win32.win32.x86_64-1.1.0.201508120742	Version 1.1.0 for Windows (64 Bit)	293.84 MB	105
amalthea-linux.gtk.x86_64-1.1.0.201508120742	Version 1.1.0 for Linux (64 Bit)	293.30 MB	26
amalthea-win32.win32.x86-1.1.0.201508120742	Version 1.1.0 for Windows (32 Bit)	293.70 MB	23
amalthea-macosx.cocoa.x86_64-1.1.0.201508120742	Version 1.1.0 for OSX (64 Bit)	293.01 MB	1
amalthea-win32.win32.x86_64-1.0.3.201406160730	Version 1.0.3 for Windows (64 Bit)	299.43 MB	673
amalthea-win32.win32.x86-1.0.3.201406160730	Version 1.0.3 for Windows (32 Bit)	299.29 MB	46
amalthea-linux.gtk.x86_64-1.0.3.201406160730	Version 1.0.3 for Linux (64 Bit)	299.04 MB	155
amalthea-macosx.cocoa.x86_64-1.0.3.201406160730	Version 1.0.3 for OSX (64 Bit)	298.94 MB	56

The official update site for the AMALTHEA platform is located at <http://platform.amalthea-project.org/update/>

Release 1.1.1

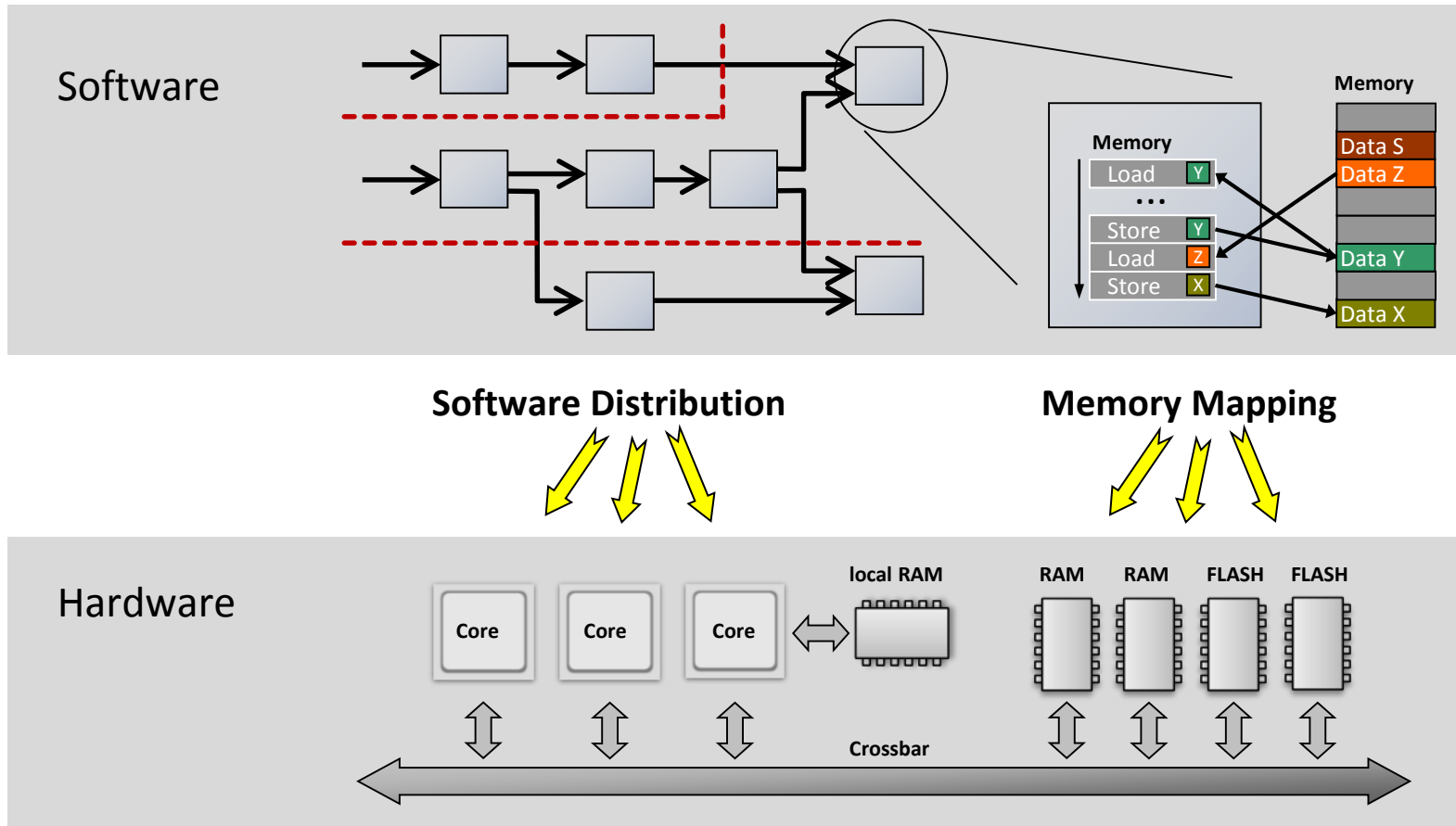
October 2015

<http://www.amalthea-project.org>

APP4MC – Application Platform Project for MultiCore

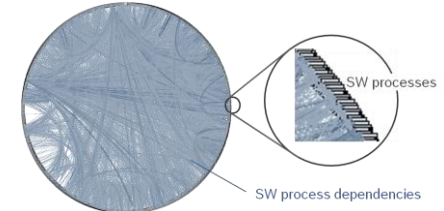
- AMALTHEA - Timeline and current project(s)
- **Challenges for embedded multi- and many-core systems**
- The AMALTHEA Platform
- Demo / Screenshots of current release
- APP4MC - Next steps

Challenges of Embedded Multi-Core



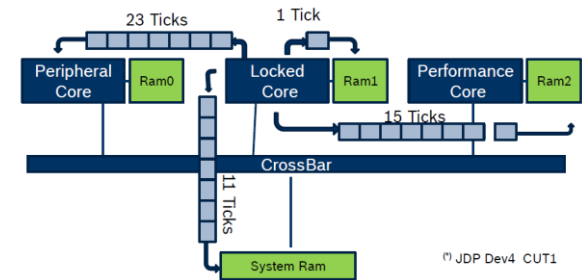
- **Typical Paradigm of Single-Core Software**

- 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

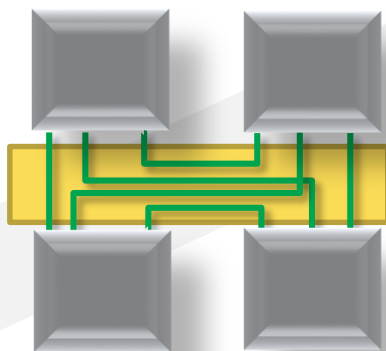


- Sophisticated new tooling required for task distribution, memory location optimization and performance analysis

Cross-core communication is a new resource bottleneck

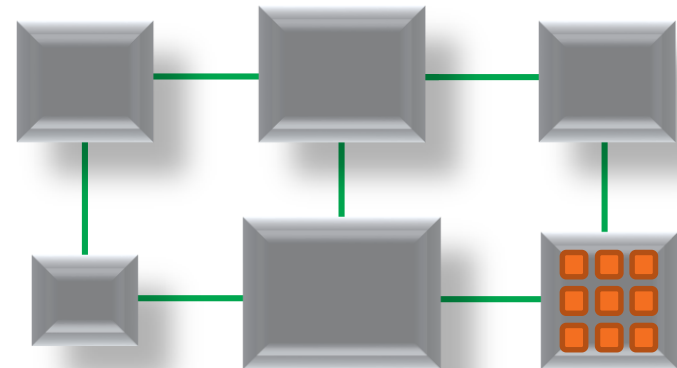
Multi-Core

- 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

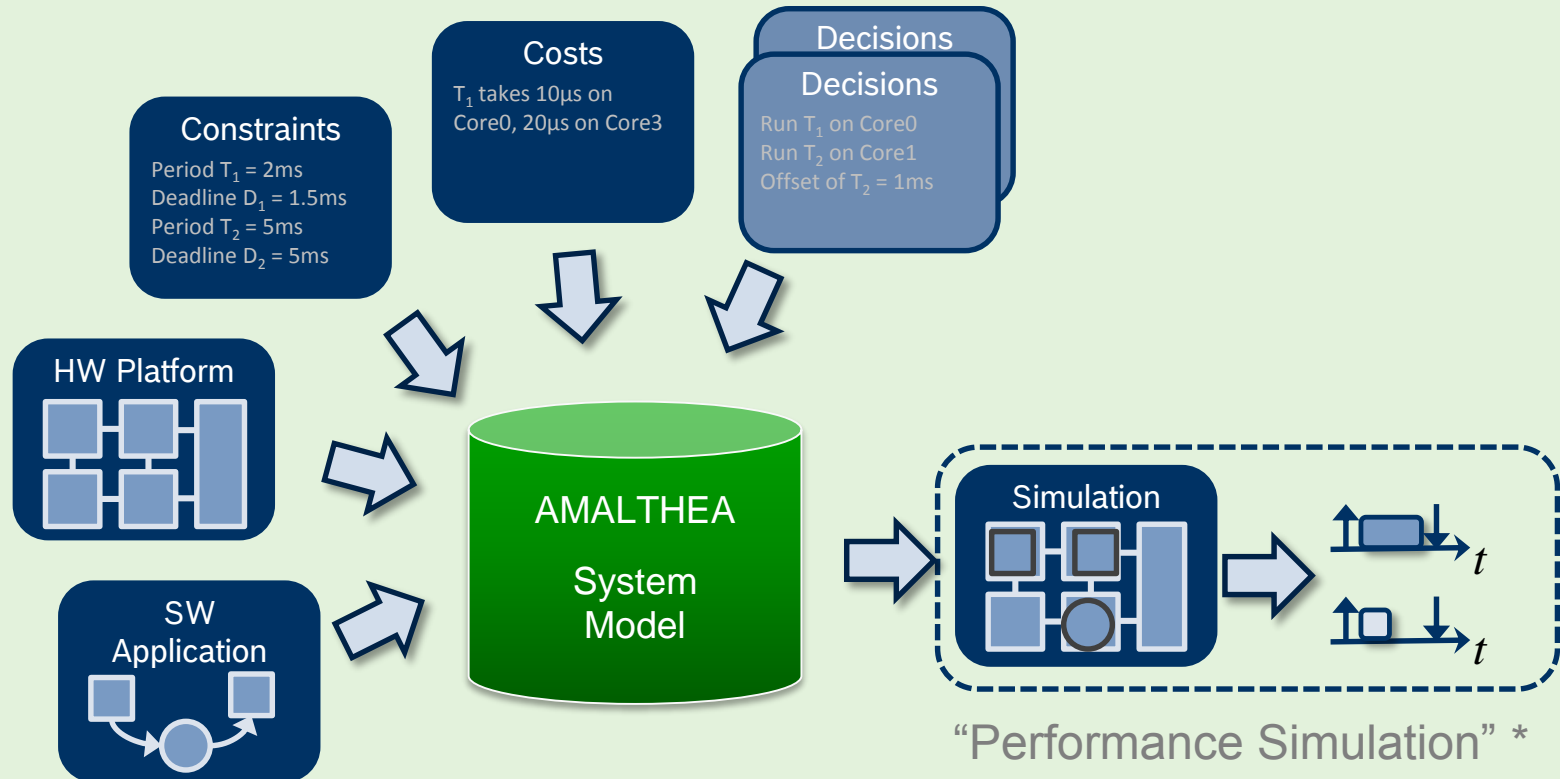
APP4MC – Application Platform Project for MultiCore

- 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

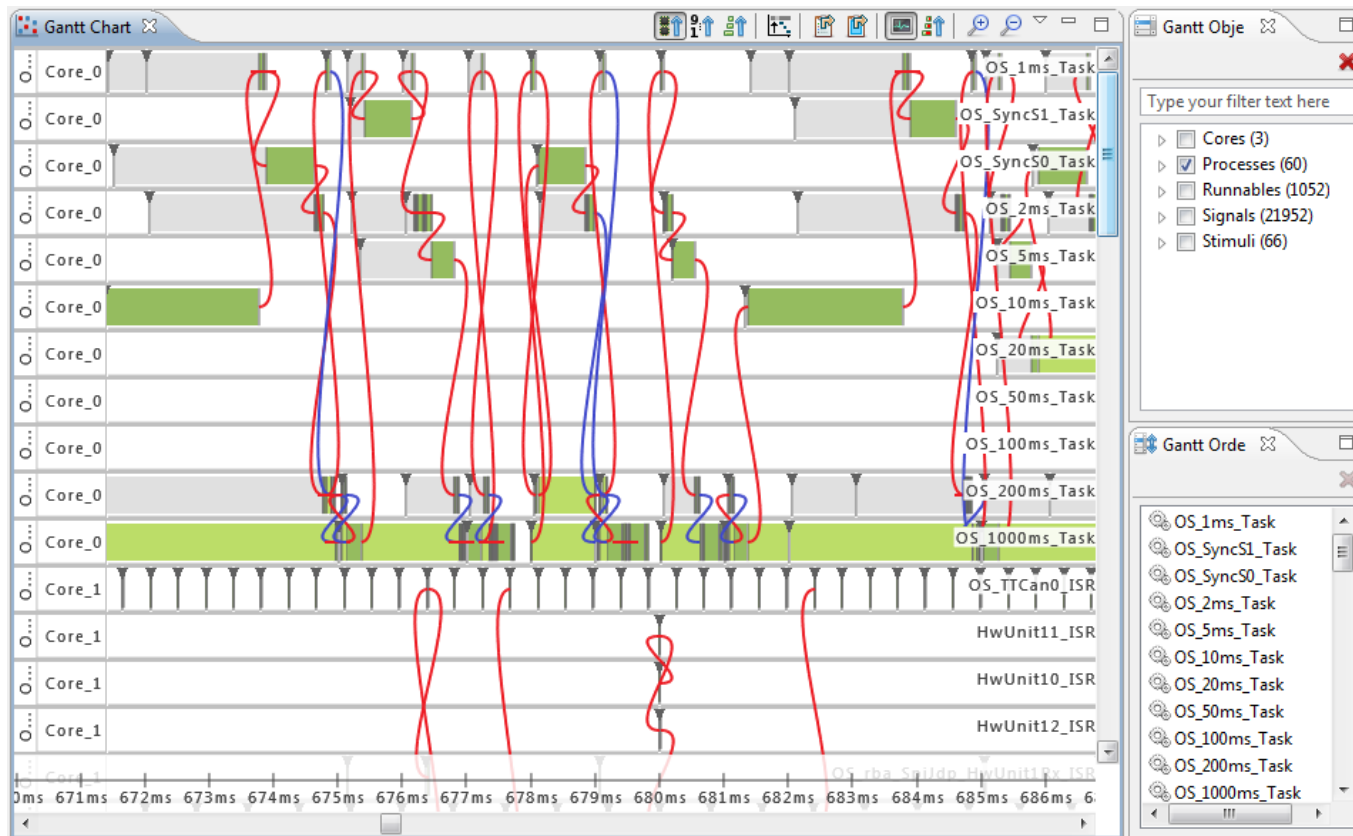
Tool platform AMALTHEA

Processing, Simulation and Analysis

AMALTHEA4public



* Focus on Timing, Scheduling

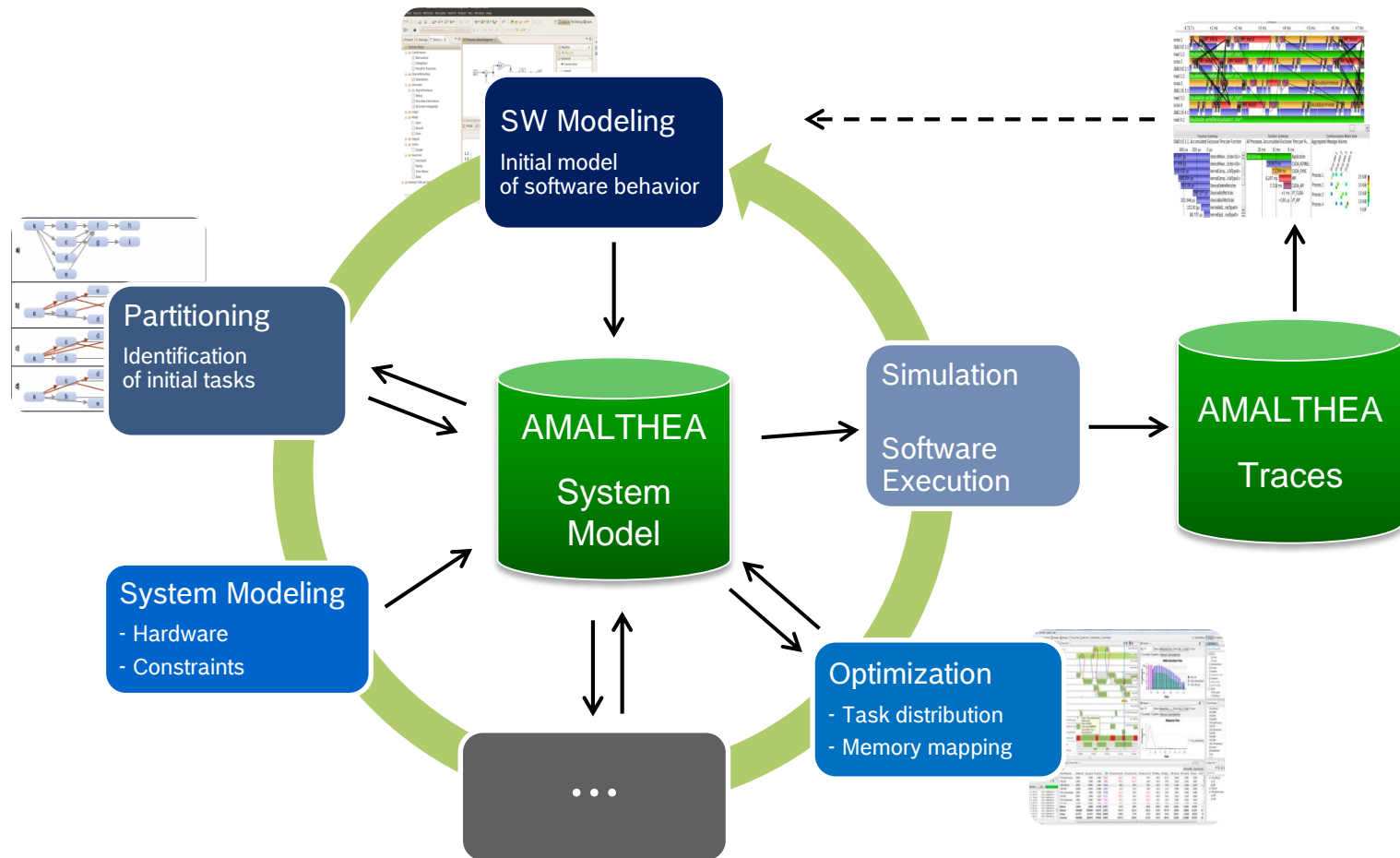


Example of a timing / scheduling simulation*

*Commercial tool – not part of the open source project

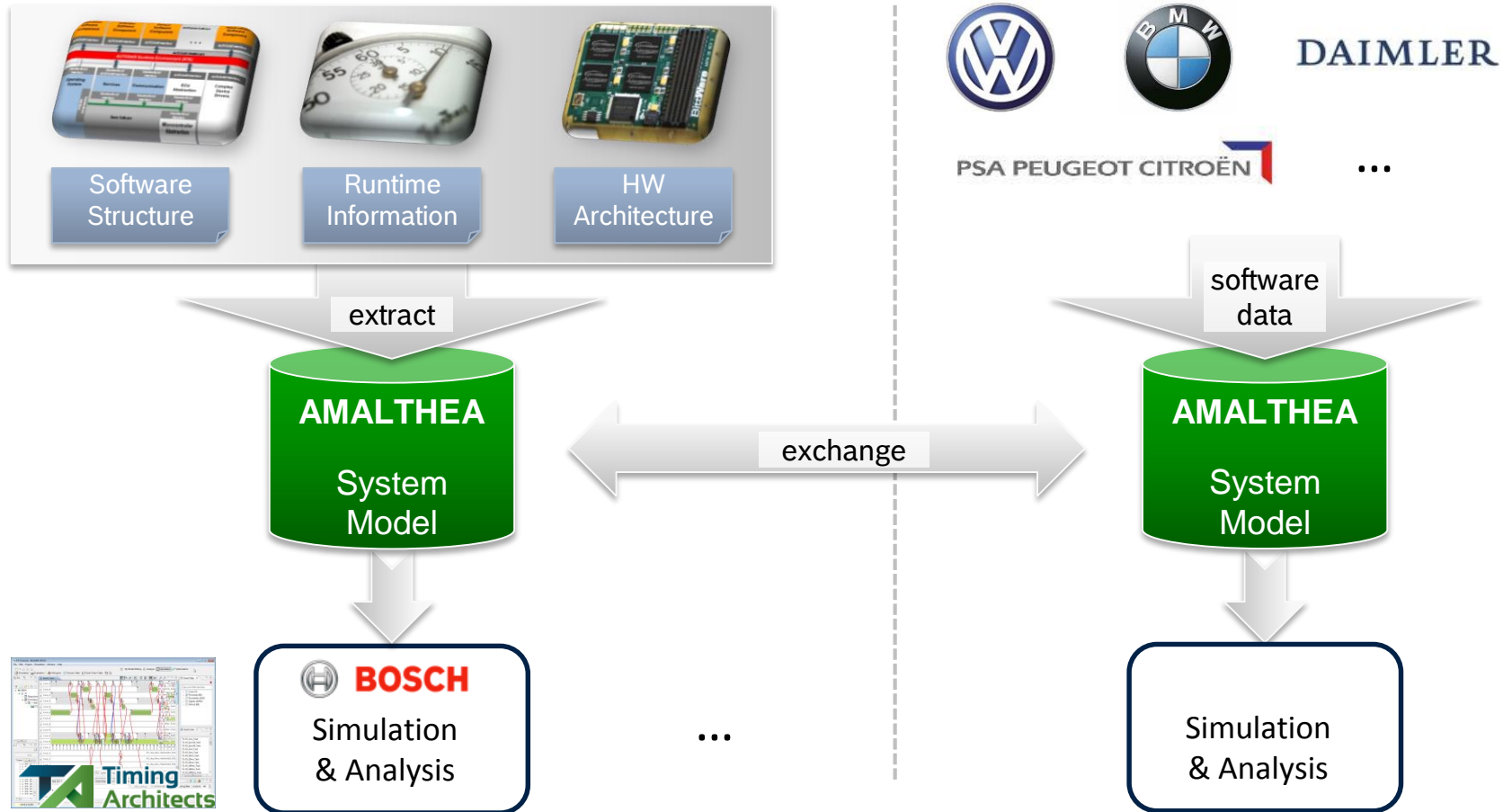
Tool platform AMALTHEA

Processing, Simulation and Analysis



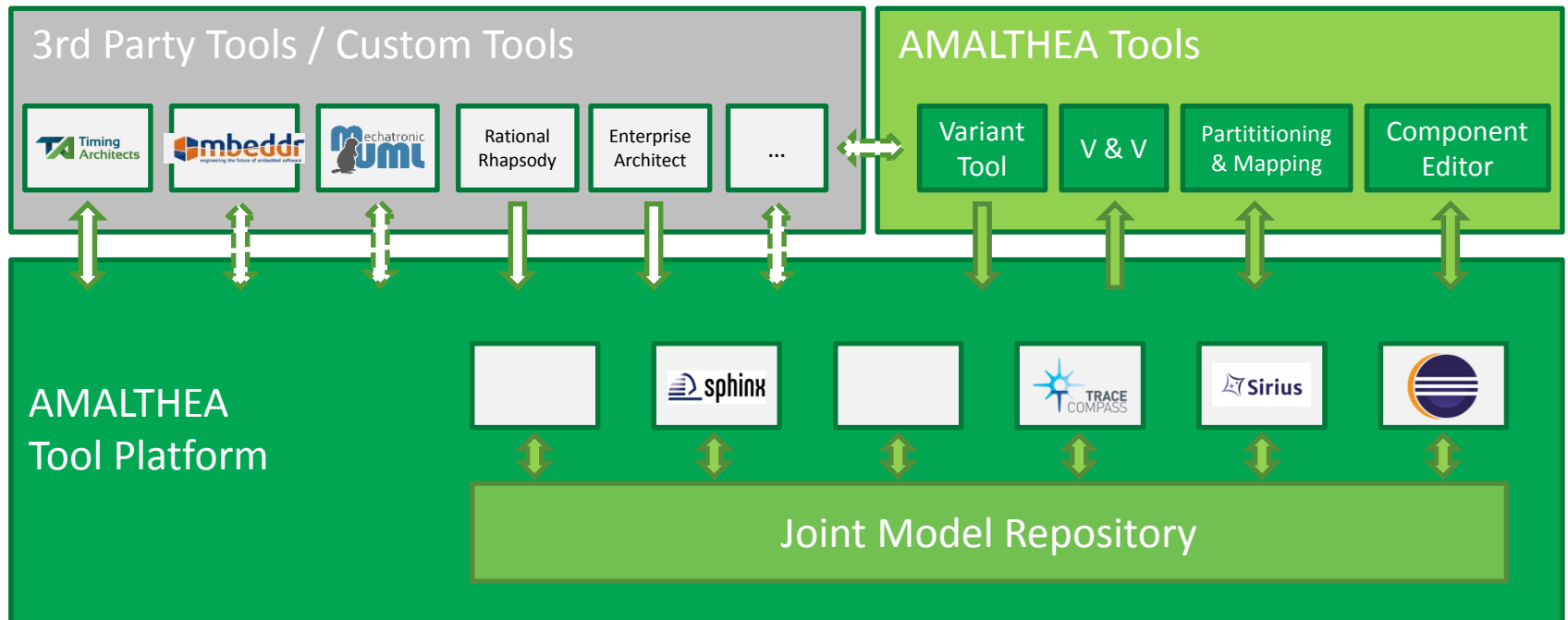
Tool platform AMALTHEA

Use cases @ BOSCH



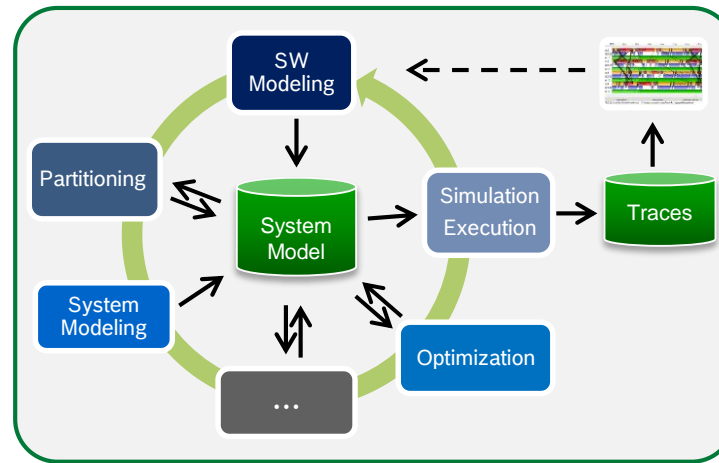
Tool platform AMALTHEA

Platform Architecture



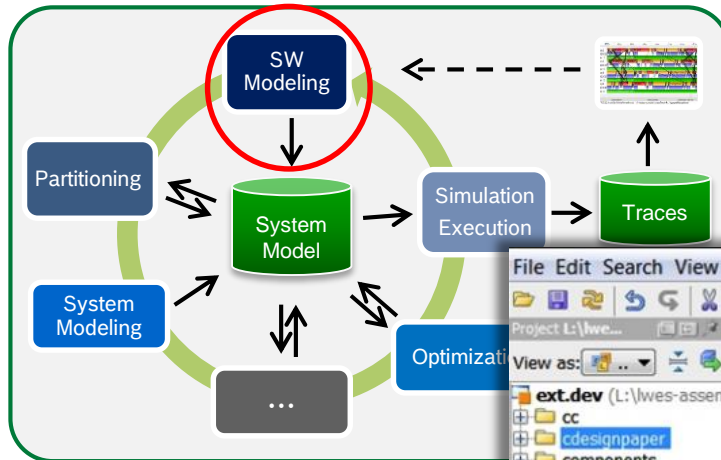
Tool platform AMALTHEA

Connection to other Eclipse Projects



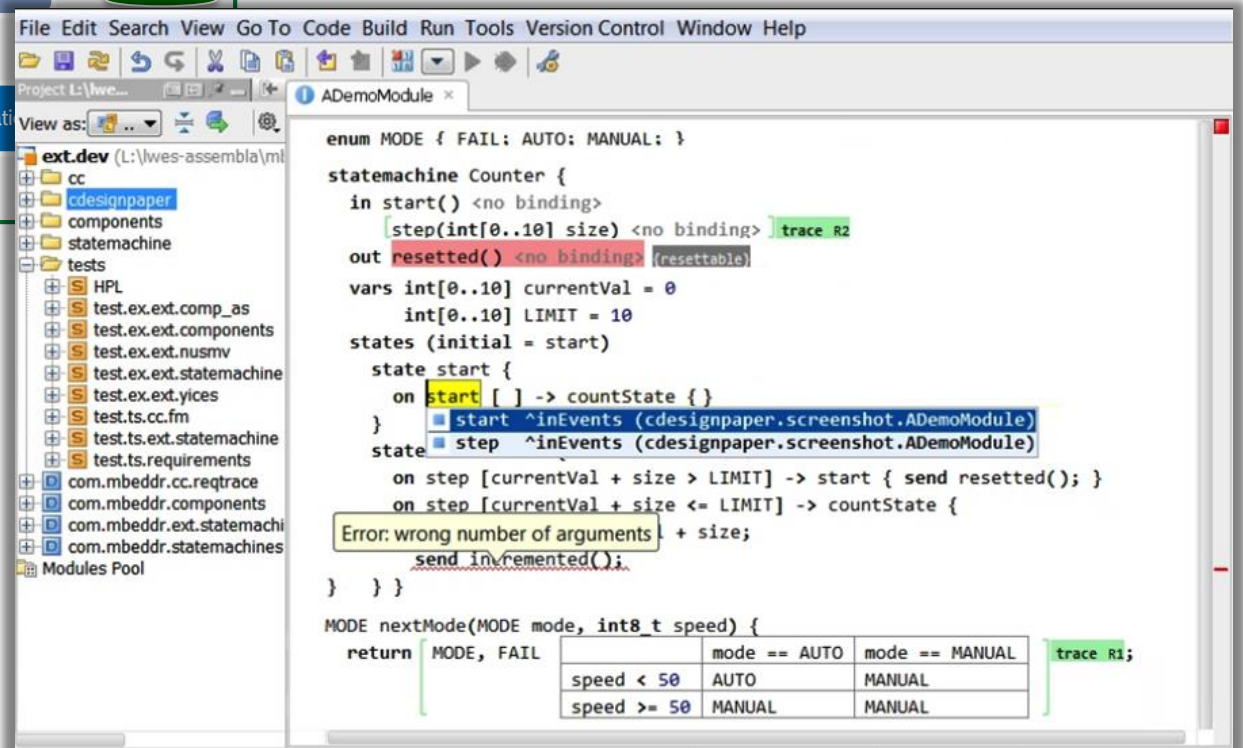
Tool platform AMALTHEA

Connection to other Eclipse Projects



planned extension

- Timing Annotations



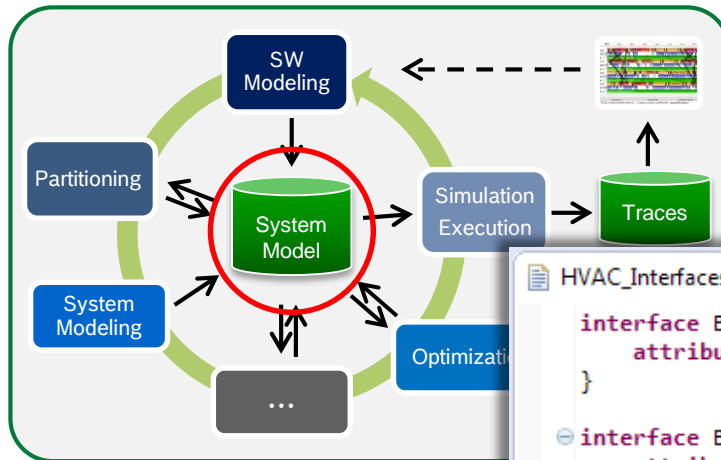
```
enum MODE { FAIL; AUTO; MANUAL; }

statemachine Counter {
  in start() <no binding>
  step(int[0..10] size) <no binding> trace R2
  out resetted() <no binding> (resettable)
  vars int[0..10] currentVal = 0
  int[0..10] LIMIT = 10
  states (initial = start)
    state start {
      on start [ ] -> countState {
        start ^inEvents (cdesignpaper.screenshot.ADemoModule)
        step ^inEvents (cdesignpaper.screenshot.ADemoModule)
      }
    }
    state
      on step [currentVal + size > LIMIT] -> start { send resetted(); }
      on step [currentVal + size <= LIMIT] -> countState {
        Error: wrong number of arguments + size;
        send incremented();
      }
  } }

MODE nextMode(MODE mode, int8 t speed) {
  return [ MODE, FAIL
    mode == AUTO mode == MANUAL
    speed < 50    AUTO      MANUAL
    speed >= 50   MANUAL    MANUAL ] trace R1;
```

Tool platform AMALTHEA

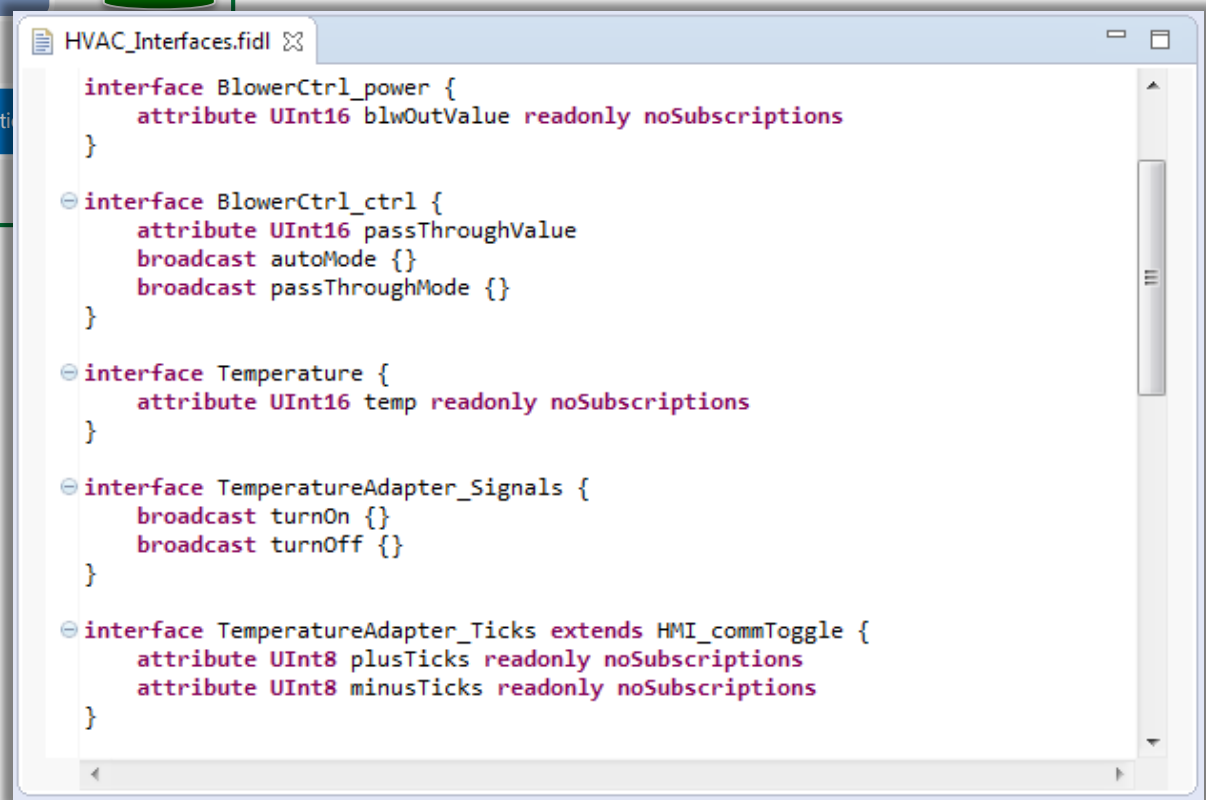
Connection to other Eclipse Projects



Franca

to describe

- Interfaces



```
interface BlowerCtrl_power {
    attribute UInt16 blwOutValue readonly noSubscriptions
}

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

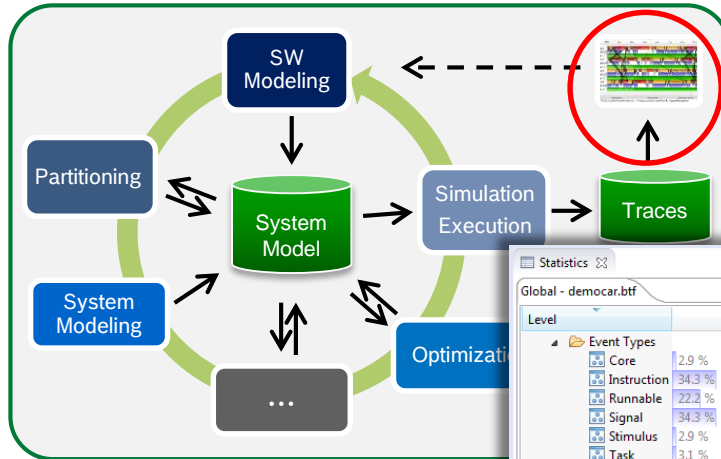
interface Temperature {
    attribute UInt16 temp readonly noSubscriptions
}

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

interface TemperatureAdapter_Ticks extends HMI_commToggle {
    attribute UInt8 plusTicks readonly noSubscriptions
    attribute UInt8 minusTicks readonly noSubscriptions
}
```

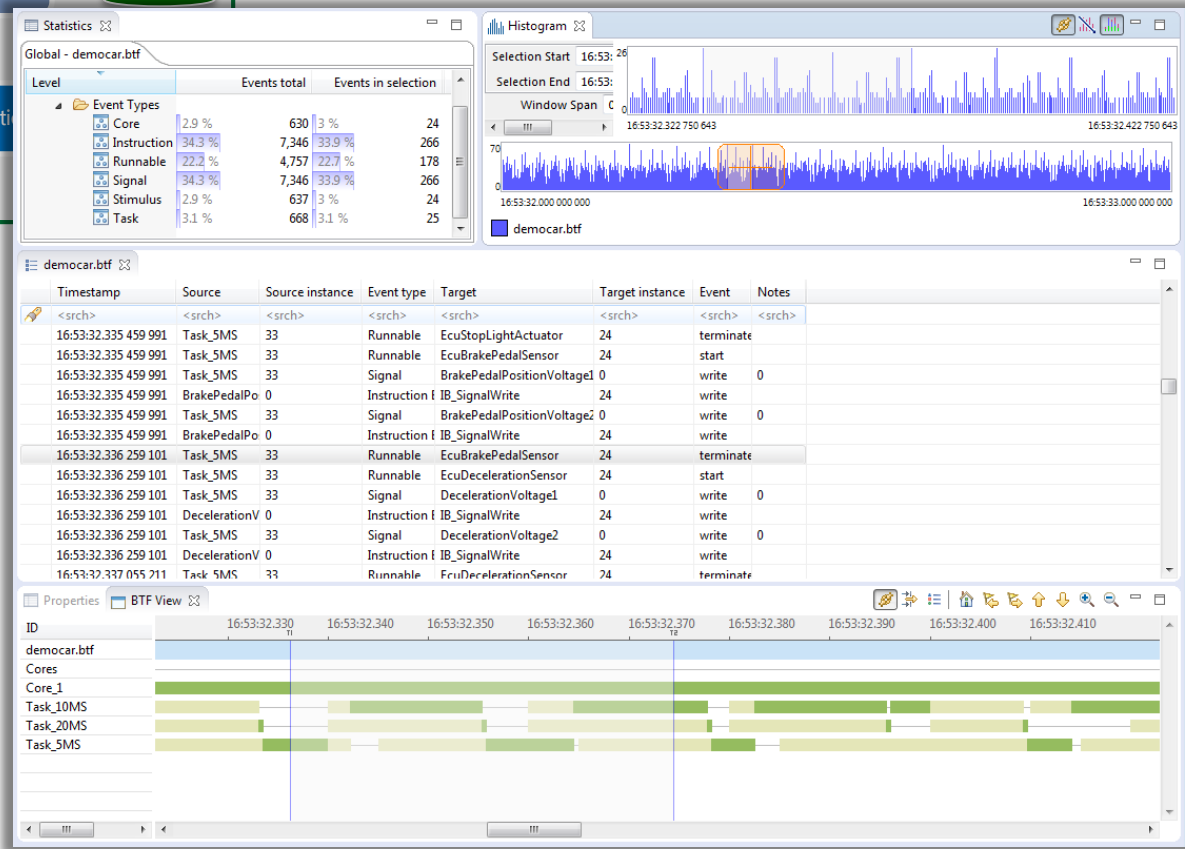
Tool platform AMALTHEA

Connection to other Eclipse Projects



implemented

- BTF Import
- BTF View



- **Eclipse IDE for Automotive Software Developers**

- **Xcore**



-> *Definition of the AMALTHEA Ecore model*

- **Sphinx**



-> *Workspace Handling, Validation, Tree Editor*

- **Xtend2**



-> *Code generation, model transformation*

- **Lyo**



-> OSLC

(separate Java libraries, no bundles, not included in orbit)

- **Ecore Tools**

EcoreTools

-> Class diagrams

(only limited connection to Xcore models)

- **Sirius**



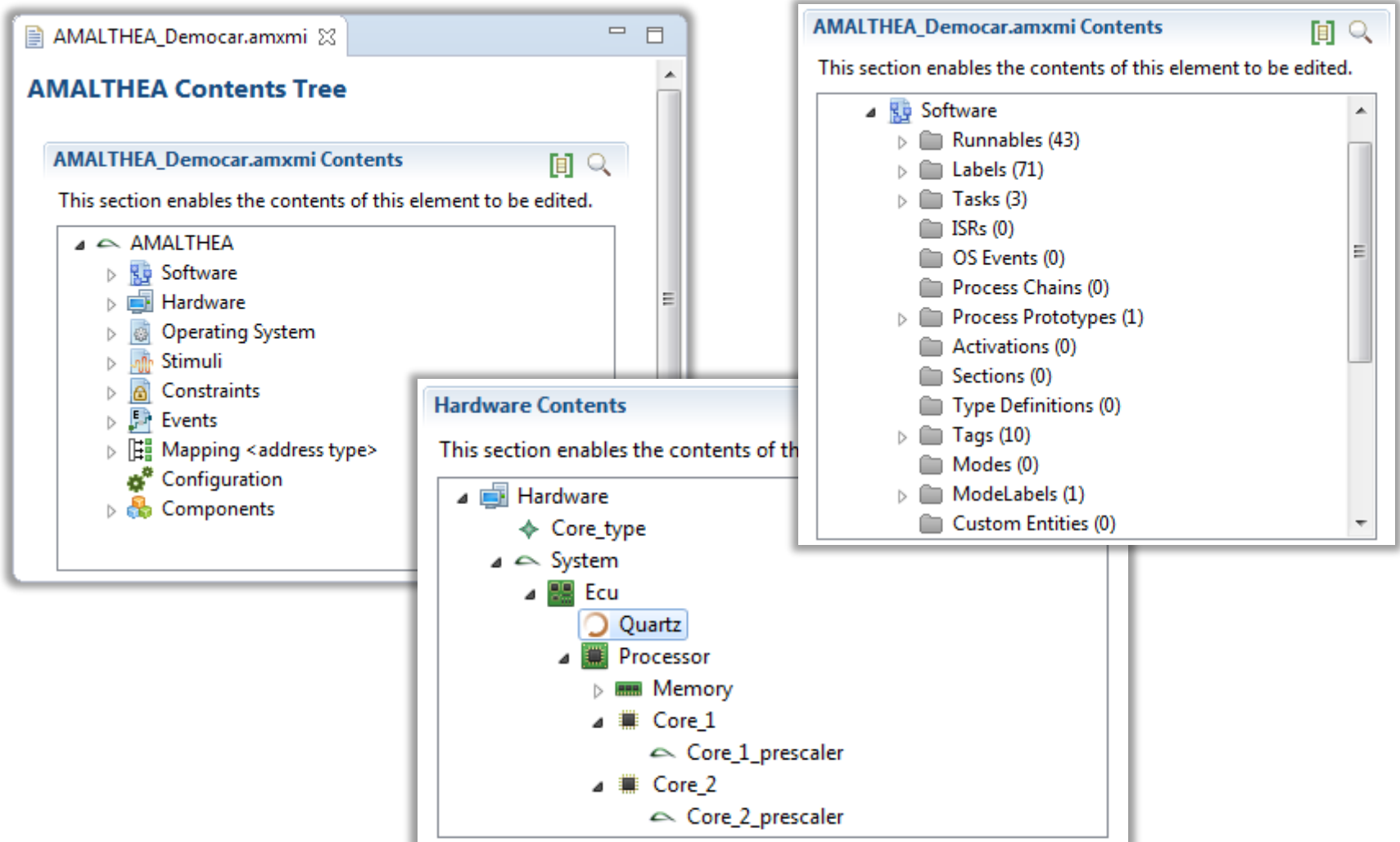
-> Graphical editors (first prototype)

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

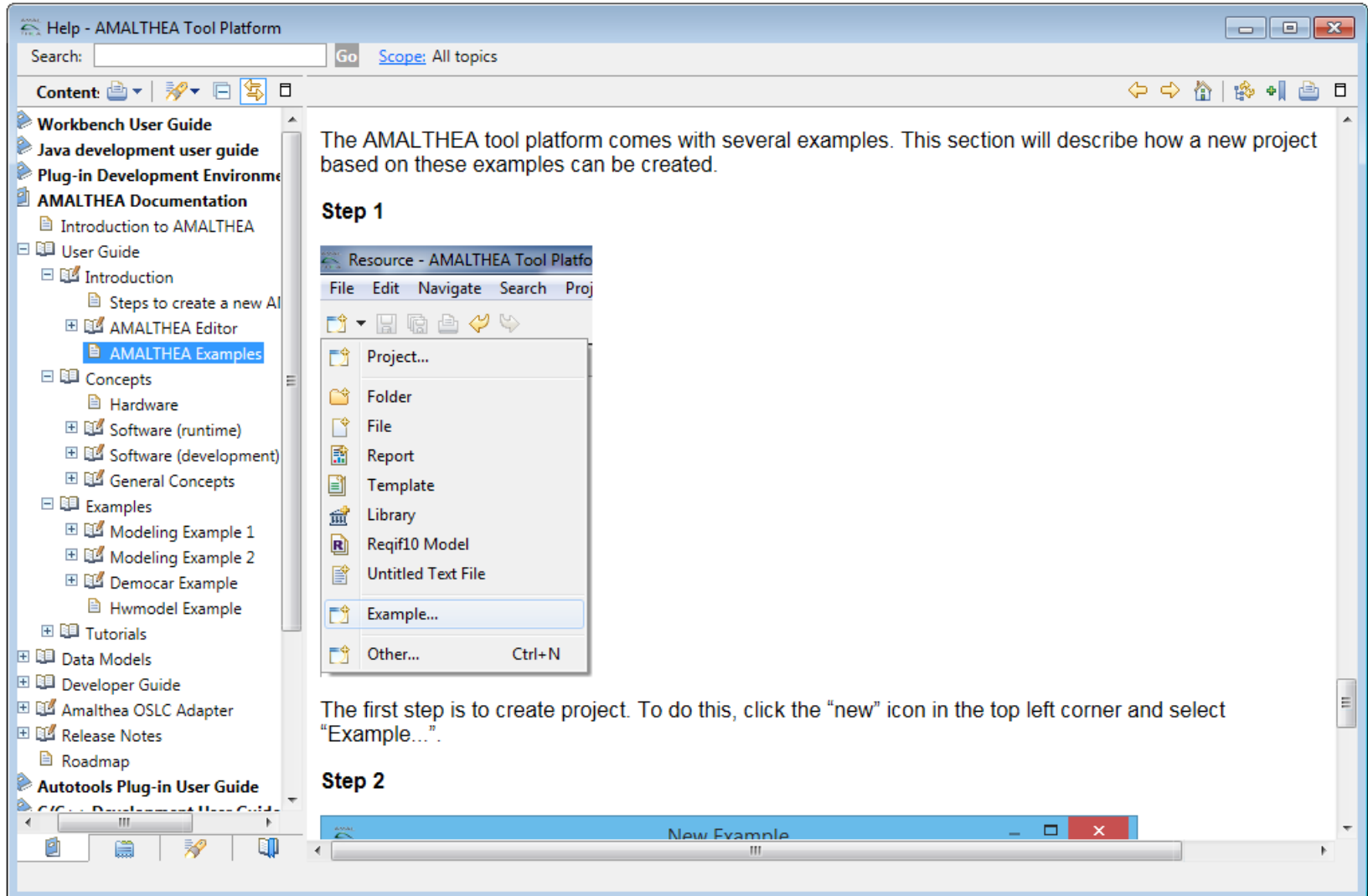


The screenshot displays the AMALTHEA Model Editor interface for the file `AMALTHEA_Democar.amxmi`. It features three overlapping panels:

- AMALTHEA Contents Tree:** A hierarchical view of the model's structure.
 - AMALTHEA
 - Software
 - Hardware
 - Operating System
 - Stimuli
 - Constraints
 - Events
 - Mapping <address type>
 - Configuration
 - Components
- AMALTHEA_Democar.amxmi Contents:** A panel for editing the contents of the selected element, showing a list of software components with their counts in parentheses.
 - Software
 - Runnables (43)
 - Labels (71)
 - Tasks (3)
 - ISRs (0)
 - OS Events (0)
 - Process Chains (0)
 - Process Prototypes (1)
 - Activations (0)
 - Sections (0)
 - Type Definitions (0)
 - Tags (10)
 - Modes (0)
 - ModelLabels (1)
 - Custom Entities (0)
- Hardware Contents:** A panel for editing the hardware components, showing a detailed tree structure.
 - Hardware
 - Core_type
 - System
 - Ecu
 - Quartz
 - Processor
 - Memory
 - Core_1
 - Core_1_prescaler
 - Core_2
 - Core_2_prescaler





Platform Demo

Eclipse Help – Examples



Help - AMALTHEA Tool Platform

Search: Go [Scope: All topics](#)

Content:    





- Workbench User Guide
- Java development user guide
- Plug-in Development Environment
- AMALTHEA Documentation
 - Introduction to AMALTHEA
 - User Guide
 - Introduction
 - Steps to create a new AMALTHEA project
 - AMALTHEA Editor
 - AMALTHEA Examples**
 - Concepts
 - Hardware
 - Software (runtime)
 - Software (development)
 - General Concepts
 - Examples
 - Modeling Example 1
 - Modeling Example 2
 - Democar Example
 - Hwmodel Example
 - Tutorials
- Data Models
- Developer Guide
- Amalthea OSLC Adapter
- Release Notes
- Roadmap
- Autotools Plug-in User Guide

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

Resource - AMALTHEA Tool Platform

File Edit Navigate Search Proj

- Project...
- Folder
- File
- Report
- Template
- Library
- Reqif10 Model
- Untitled Text File
- Example...**
- Other... Ctrl+N

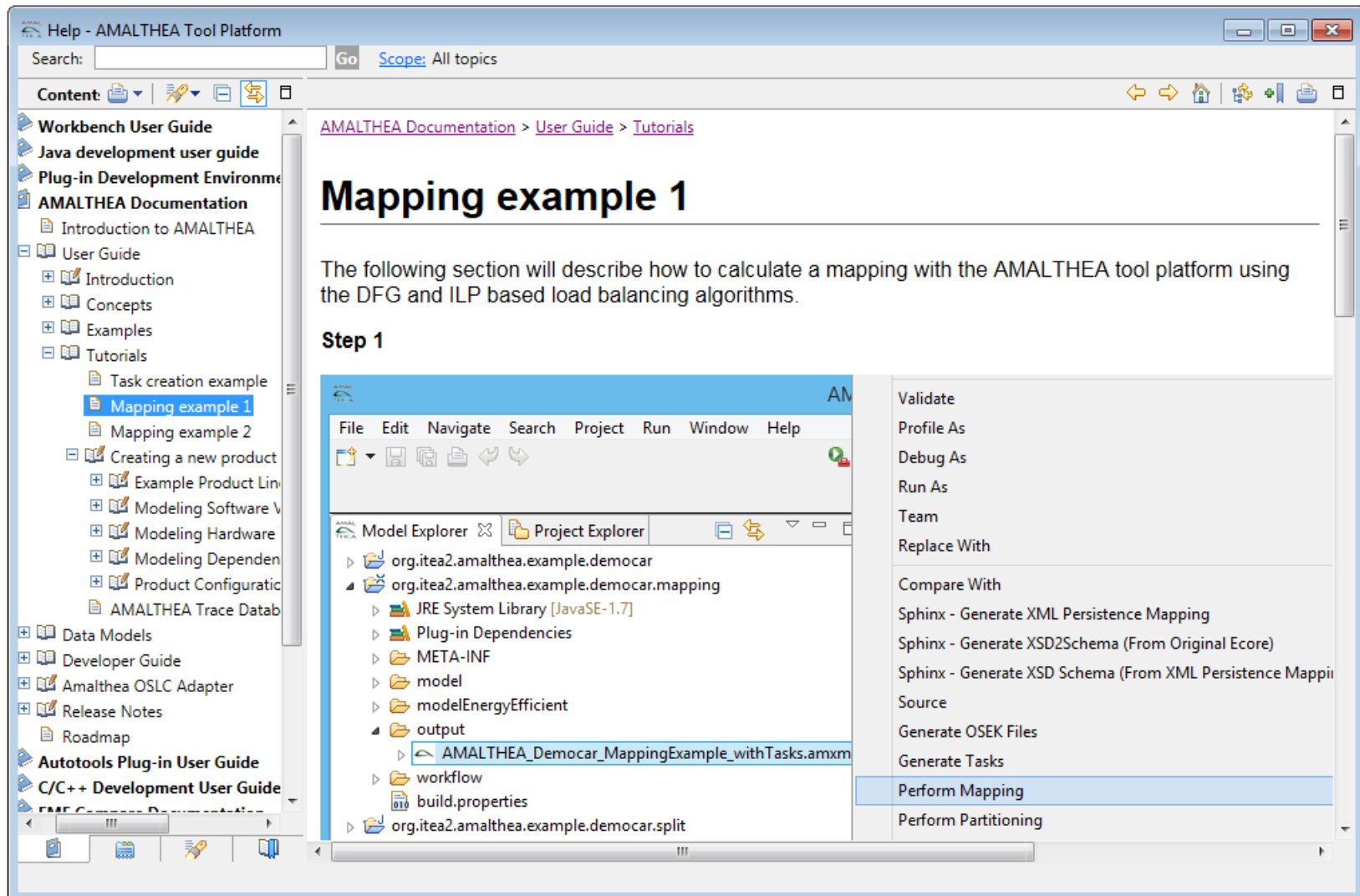
The first step is to create project. To do this, click the “new” icon in the top left corner and select “Example...”.

Step 2

New Example

Platform Demo

Eclipse Help – Tutorials



The screenshot displays the Eclipse IDE interface with the AMALTHEA Help documentation open. The left sidebar shows the 'Content' tree with 'AMALTHEA Documentation' expanded, and 'User Guide' > 'Tutorials' > 'Mapping example 1' selected. The main editor area shows the 'Mapping example 1' page, which includes a breadcrumb trail 'AMALTHEA Documentation > User Guide > Tutorials', a title 'Mapping example 1', and a description: 'The following section will describe how to calculate a mapping with the AMALTHEA tool platform using the DFG and ILP based load balancing algorithms.' Below this, 'Step 1' is indicated. The bottom of the IDE shows the 'Model Explorer' and 'Project Explorer' views. The 'Model Explorer' displays a project structure for 'org.itea2.amalthea.example.democar', with 'org.itea2.amalthea.example.democar.mapping' expanded, showing sub-projects like 'JRE System Library [JavaSE-1.7]', 'Plug-in Dependencies', 'META-INF', 'model', 'modelEnergyEfficient', 'output' (containing 'AMALTHEA_Democar_MappingExample_withTasks.amxm'), 'workflow', 'build.properties', and 'org.itea2.amalthea.example.democar.split'. The 'Project Explorer' view is also visible. On the right, a context menu is open, listing various actions such as 'Validate', 'Profile As', 'Debug As', 'Run As', 'Team', 'Replace With', 'Compare With', 'Sphinx - Generate XML Persistence Mapping', 'Sphinx - Generate XSD2Schema (From Original Ecore)', 'Sphinx - Generate XSD Schema (From XML Persistence Mapping)', 'Source', 'Generate OSEK Files', 'Generate Tasks', 'Perform Mapping' (highlighted), and 'Perform Partitioning'.

Platform Demo

Eclipse Help – Data Models

Help - AMALTHEA Tool Platform

Search: Go Scope: All topics

Content:

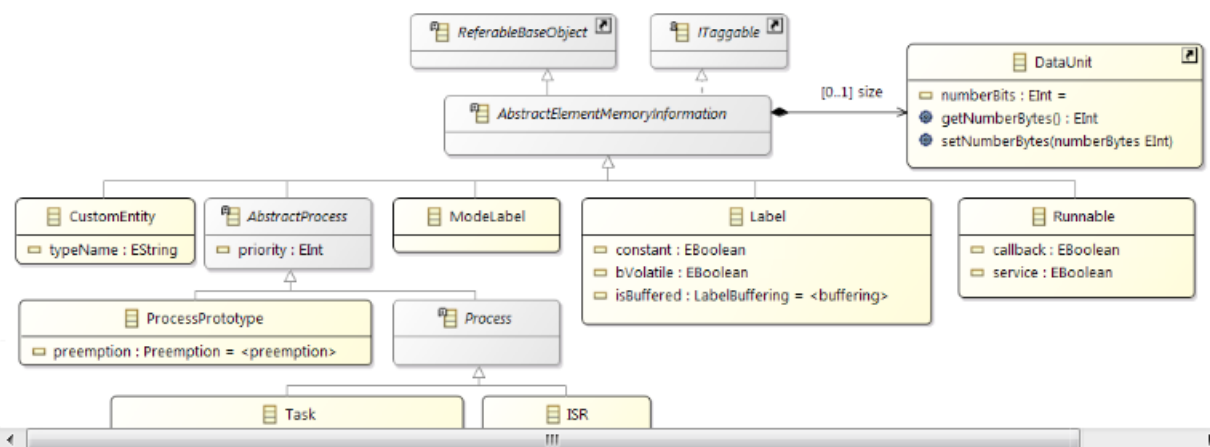
- Workbench User Guide
- Java development user guide
- Plug-in Development Enviror
- AMALTHEA Documentation
 - Introduction to AMALTHEA
 - User Guide
 - Data Models
 - Data Model Overview
 - Hardware model
 - Common Model
 - Configuration Model
 - Constraints Model
 - Event Model
 - Mapping Model
 - OS Model
 - PropertyConstraints Mo
 - Stimuli Model
 - Software Model
 - Components Model
 - Developer Guide
 - Amalthea OSLC Adapter
 - Release Notes
 - Roadmap
- Autotools Plug-in User Guide
- C/C++ Development User Gu
- EMF Compare Documentatio
- Franca User Guide
- GNU Tool On-Line Docu

Software Model

The AMALTHEA software model is central accessible through the *SWModel* 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):



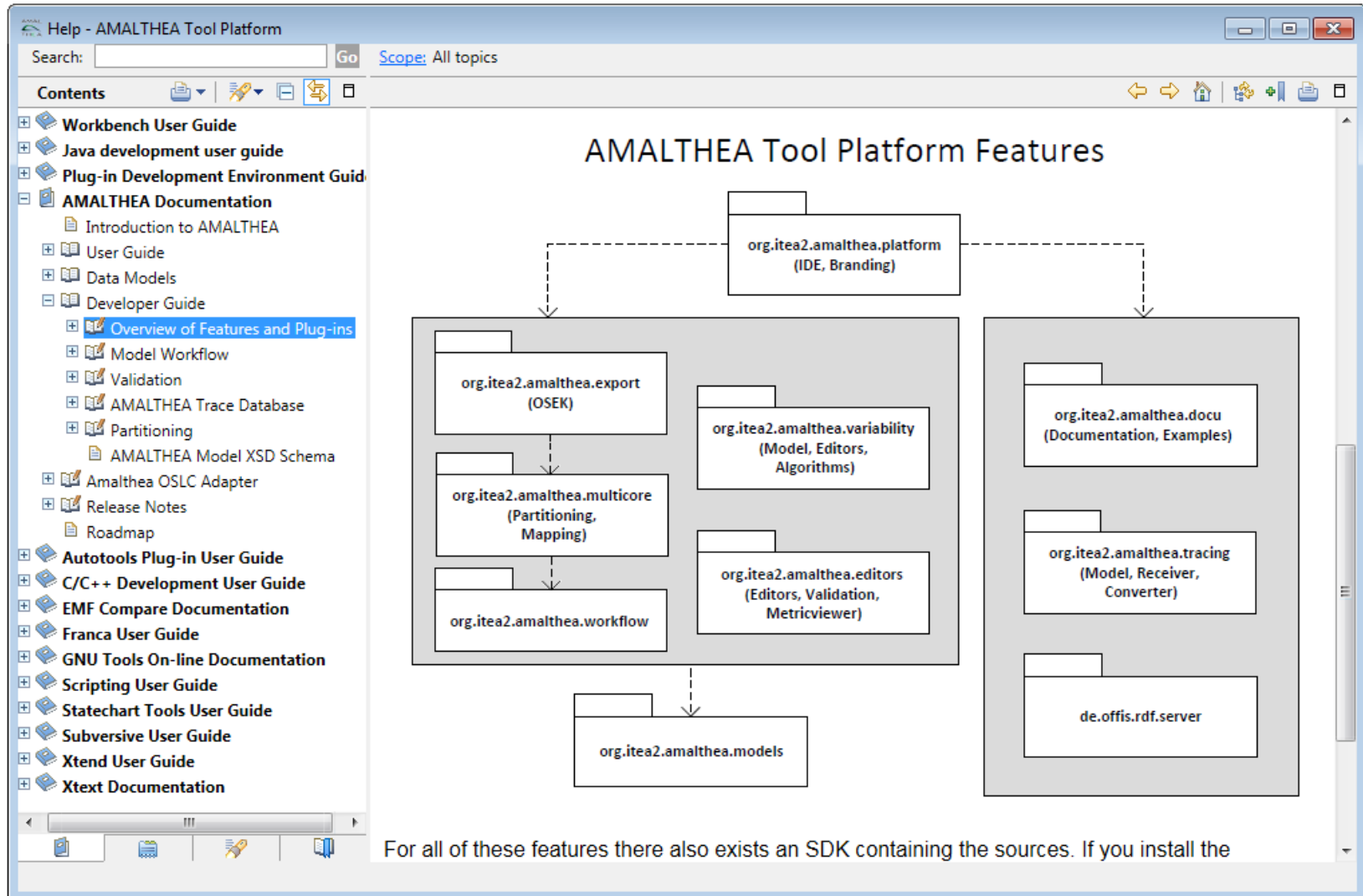
```
classDiagram
    class ReferableBaseObject
    class ITaggable
    class AbstractElementMemoryInformation
    class CustomEntity
    class AbstractProcess
    class ModelLabel
    class Label
    class Runnable
    class DataUnit
    class ProcessPrototype
    class Process
    class Task
    class ISR

    ReferableBaseObject <|-- AbstractElementMemoryInformation
    ITaggable <|-- AbstractElementMemoryInformation
    AbstractElementMemoryInformation <|-- CustomEntity
    AbstractElementMemoryInformation <|-- AbstractProcess
    AbstractElementMemoryInformation <|-- ModelLabel
    AbstractElementMemoryInformation <|-- Label
    AbstractElementMemoryInformation <|-- Runnable
    AbstractProcess <|-- ProcessPrototype
    ModelLabel <|-- Process
    ProcessPrototype <|-- Task
    Process <|-- ISR

    CustomEntity "1" -- "1" type : EString
    AbstractProcess "1" -- "1" priority : EInt
    ModelLabel "1" -- "1" preemption : Preemption
    ProcessPrototype "1" -- "1" preemption : Preemption
    Process "1" -- "1" preemption : Preemption
    Label "1" -- "1" constant : EBoolean
    Label "1" -- "1" bVolatile : EBoolean
    Label "1" -- "1" isBuffered : LabelBuffering
    Runnable "1" -- "1" callback : EBoolean
    Runnable "1" -- "1" service : EBoolean
    DataUnit "1" -- "1" numberBits : EInt
    DataUnit "1" -- "1" getNumberBytes() : EInt
    DataUnit "1" -- "1" setNumberBytes(numberBytes EInt)
```

Platform Demo

Eclipse Help – Developer Guide



The screenshot shows the Eclipse Help - AMALTHEA Tool Platform window. The left sidebar contains a 'Contents' tree with the following items:

- Workbench User Guide
- Java development user guide
- Plug-in Development Environment Guide
- AMALTHEA Documentation
 - Introduction to AMALTHEA
 - User Guide
 - Data Models
 - Developer Guide
 - Overview of Features and Plug-ins**
 - Model Workflow
 - Validation
 - AMALTHEA Trace Database
 - Partitioning
 - AMALTHEA Model XSD Schema
 - Amalthea OSLC Adapter
 - Release Notes
 - Roadmap
- Autotools Plug-in User Guide
- C/C++ Development User Guide
- EMF Compare Documentation
- Franca User Guide
- GNU Tools On-line Documentation
- Scripting User Guide
- Statechart Tools User Guide
- Subversive User Guide
- Xtend User Guide
- Xtext Documentation

The main content area displays the 'AMALTHEA Tool Platform Features' diagram. The diagram shows the following components and their relationships:

- org.itea2.amalthea.platform (IDE, Branding)** (Top)
- org.itea2.amalthea.export (OSEK)** (Left, connected to platform by a dashed arrow)
- org.itea2.amalthea.multicore (Partitioning, Mapping)** (Left, connected to export by a dashed arrow)
- org.itea2.amalthea.workflow** (Left, connected to multicore by a dashed arrow)
- org.itea2.amalthea.variability (Model, Editors, Algorithms)** (Right, connected to platform by a dashed arrow)
- org.itea2.amalthea.editors (Editors, Validation, Metricviewer)** (Right, connected to variability by a dashed arrow)
- org.itea2.amalthea.docu (Documentation, Examples)** (Right, connected to platform by a dashed arrow)
- org.itea2.amalthea.tracing (Model, Receiver, Converter)** (Right, connected to docu by a dashed arrow)
- de.offis.rdf.server** (Right, connected to tracing by a dashed arrow)
- org.itea2.amalthea.models** (Bottom, connected to workflow by a dashed arrow)

For all of these features there also exists an SDK containing the sources. If you install the

Platform Demo

Eclipse Help – Developer Guide



The screenshot shows the Eclipse Help window for the AMALTHEA Tool Platform. The left sidebar contains a 'Contents' tree with the following structure:

- Workbench User Guide
- Java development user guide
- Plug-in Development Environment
- AMALTHEA Documentation
 - Introduction to AMALTHEA
 - User Guide
 - Data Models
 - Developer Guide
 - Overview of Features and Plug-ins
 - Model Workflow
 - Introduction
 - General Structure
 - Available Components
 - Reader
 - Writer
 - Add Schedule Points
 - Create Tasks
 - Generate Mapping
 - Overall Sample

The 'Adding a new workflow component' page is selected in the tree. The main content area displays the title 'Adding a new workflow component' and the text: 'Below you will find a sample how to add and implement a new workflow component.'

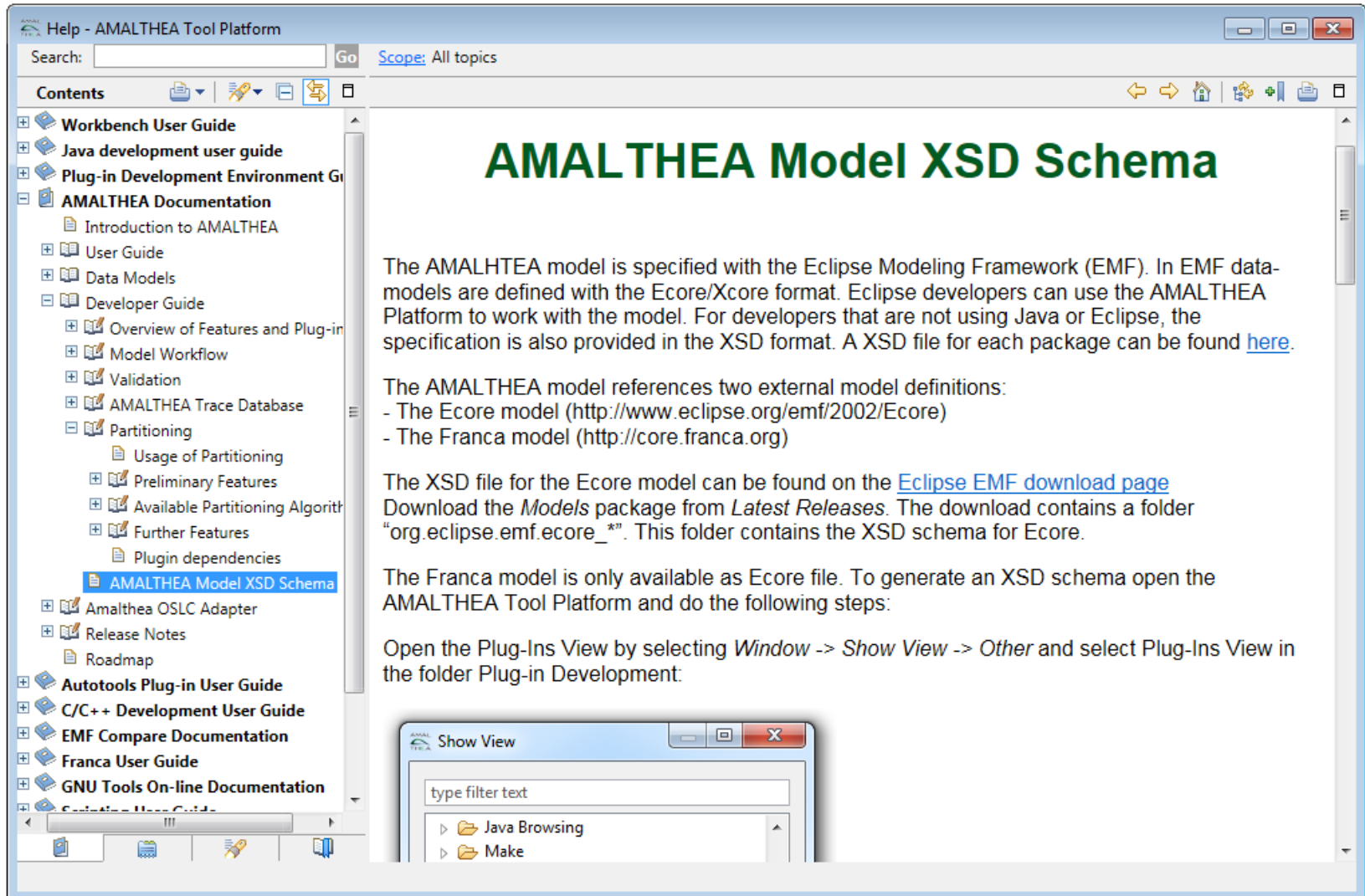
Create project

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.emf.mwe2.launch, org.eclipse.emf.mwe2.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).

```
@Override
protected void invokeInternal(final WorkflowContext ctx, final ProgressMonitor monitor, f
    // some checking if sw model is available
    if (null == getAmaltheaModel(ctx).getSwModel()) {
        issues.addError(this, "No proper SWModel available!", getModelSlot());
```

Platform Demo

Eclipse Help – Developer Guide



The screenshot shows the 'Help - AMALTHEA Tool Platform' window. The left sidebar contains a 'Contents' tree with the following structure:

- Workbench User Guide
- Java development user guide
- Plug-in Development Environment Guide
- AMALTHEA Documentation
 - Introduction to AMALTHEA
 - User Guide
 - Data Models
 - Developer Guide
 - Overview of Features and Plug-in
 - Model Workflow
 - Validation
 - AMALTHEA Trace Database
 - Partitioning
 - Usage of Partitioning
 - Preliminary Features
 - Available Partitioning Algorithms
 - Further Features
 - Plugin dependencies
 - AMALTHEA Model XSD Schema (highlighted)
 - Amalthea OSLC Adapter
 - Release Notes
 - Roadmap
- Autotools Plug-in User Guide
- C/C++ Development User Guide
- EMF Compare Documentation
- Franca User Guide
- GNU Tools On-line Documentation
- Simulation User Guide

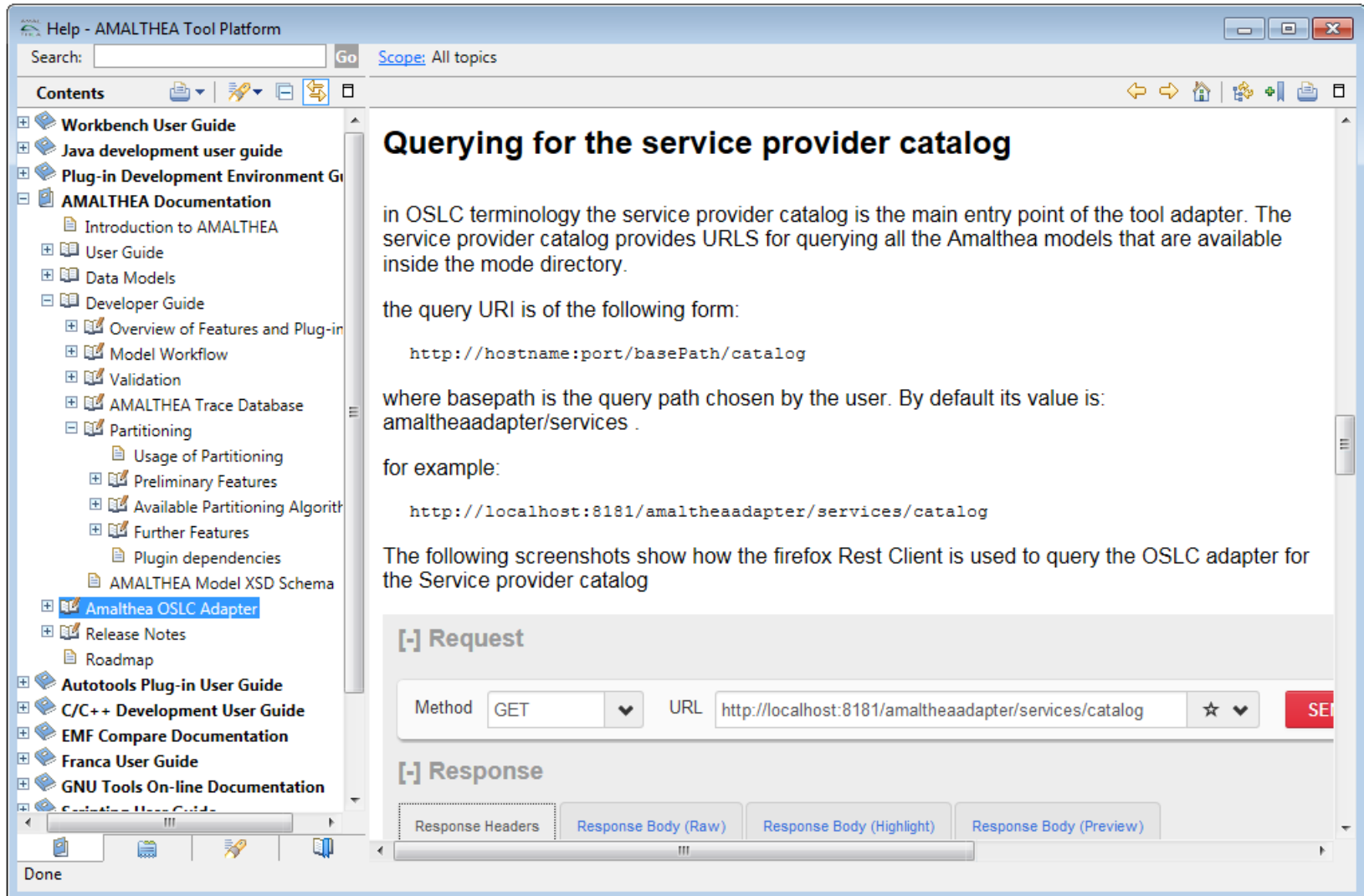
The main content area displays the 'AMALTHEA Model XSD Schema' page. The title is in large green font. The text describes the AMALTHEA model's specification using the Eclipse Modeling Framework (EMF) and Ecore/Xcore format. It mentions that for developers not using Java or Eclipse, the specification is also provided in XSD format, with a link to find XSD files. It lists two external model definitions: the Ecore model (http://www.eclipse.org/emf/2002/Ecore) and the Franca model (http://core.franca.org). It then explains how to find the XSD file for the Ecore model on the Eclipse EMF download page, noting that the download contains a folder 'org.eclipse.emf.ecore_*' which contains the XSD schema for Ecore. Finally, it states that the Franca model is only available as an Ecore file and provides steps to generate an XSD schema by opening the AMALTHEA Tool Platform and following specific steps in the Plug-Ins View.

Open the Plug-Ins View by selecting *Window -> Show View -> Other* and select Plug-Ins View in the folder Plug-in Development:

An inset window titled 'Show View' is shown at the bottom, displaying a search filter 'type filter text' and a list of views: 'Java Browsing' and 'Make'.

Platform Demo

Eclipse Help – Developer Guide



The screenshot shows the Eclipse Help - AMALTHEA Tool Platform window. The left sidebar contains a 'Contents' tree with the following structure:

- Workbench User Guide
- Java development user guide
- Plug-in Development Environment Guide
- AMALTHEA Documentation
 - Introduction to AMALTHEA
 - User Guide
 - Data Models
 - Developer Guide
 - Overview of Features and Plug-in
 - Model Workflow
 - Validation
 - AMALTHEA Trace Database
 - Partitioning
 - Usage of Partitioning
 - Preliminary Features
 - Available Partitioning Algorithms
 - Further Features
 - Plugin dependencies
 - AMALTHEA Model XSD Schema
 - Amalthea OSLC Adapter (highlighted)
 - Release Notes
 - Roadmap
- Autotools Plug-in User Guide
- C/C++ Development User Guide
- EMF Compare Documentation
- Franca User Guide
- GNU Tools On-line Documentation
- Scintilla User Guide

The main content area displays the article 'Querying for the service provider catalog'. The text reads:

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

Below the text, there is a screenshot of the Firefox Rest Client interface. It shows a 'Request' section with the following details:

- Method: GET
- URL: `http://localhost:8181/amaltheaadapter/services/catalog`

The 'Response' section is visible below the request, with tabs for 'Response Headers', 'Response Body (Raw)', 'Response Body (Highlight)', and 'Response Body (Preview)'.

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



eclipse

GETTING STARTED MEMBERS PROJECTS MORE

HOME / PROJECTS / TECHNOLOGY PROJECT / APP4MC

APP4MC

Overview Downloads Who's Involved Developer Resources Governance Contact Us

The APP4MC project provides a tool chain environment and de-facto standard to integrate tools for all major design steps in the multi- and many-core development phase. A basic set of tools will be available to demonstrate all the steps needed in the development process. Companies and R&D partners will benefit from the de-facto standard for tool chains and the support given by the features of the extended APP4MC tool chain platform. The platform can be easily adapted with commercial or in-house tools. This eases up the cooperation between different companies and organisations where various tools are already in place but based on different data inputs. These hassles will be eliminated by the APP4MC system model as a central element for the entire tool chain.

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

HW Platform

Application

AMALTHEA System Model

Simulation

Licenses:
Eclipse Public License 1.0

RELATED PROJECTS

Project Hierarchy:

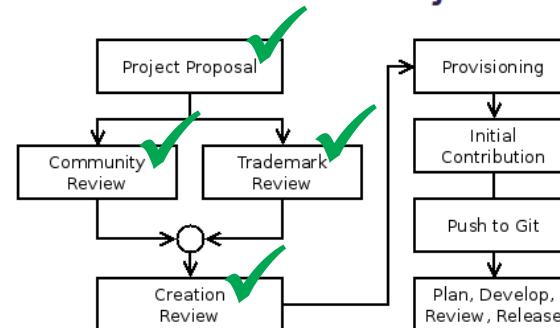
- » Technology Project
- » APP4MC

<http://projects.eclipse.org/projects/technology.app4mc>



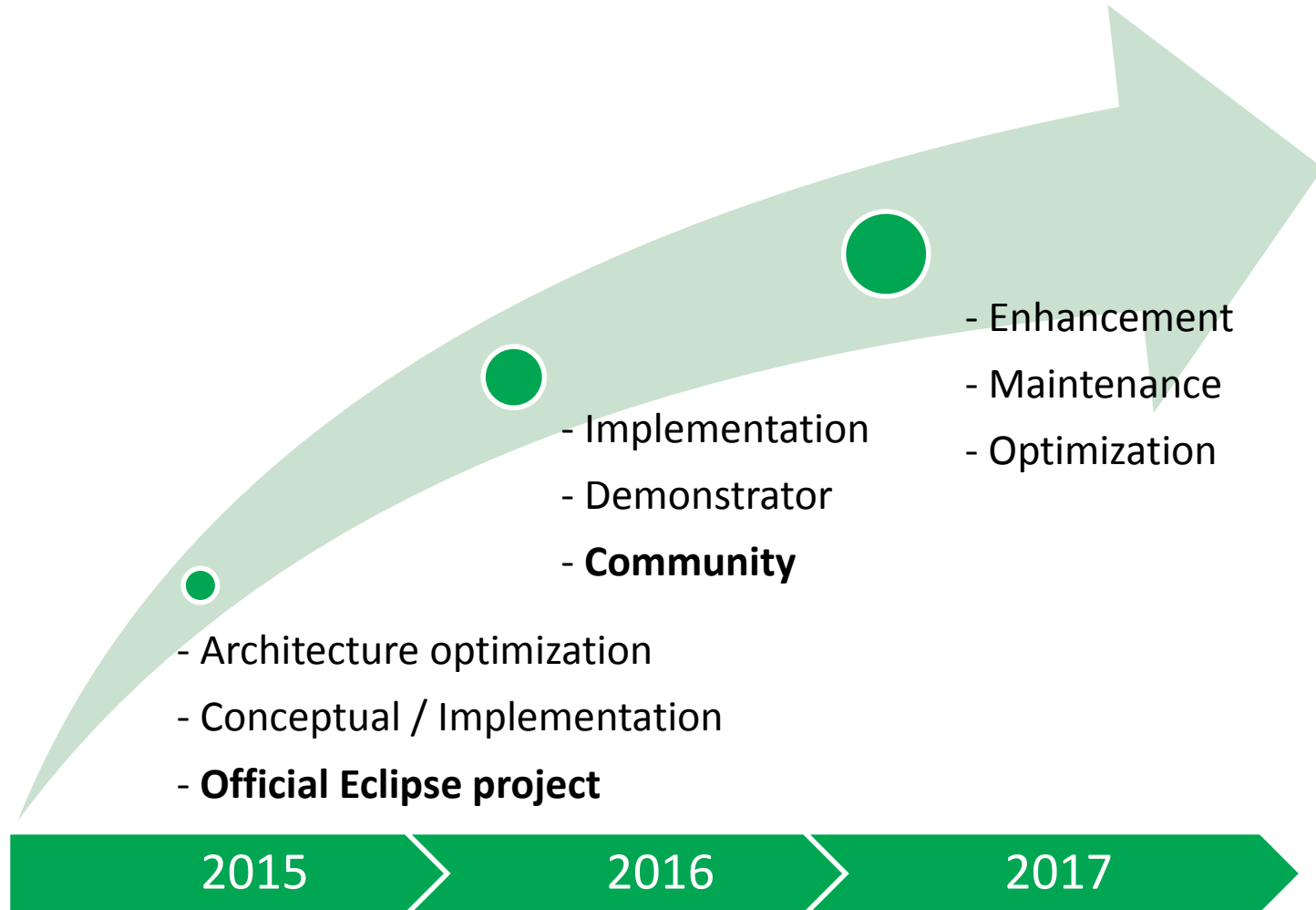
- Open Source **Eclipse project** is created
- Current work: Committer Agreements, Code refactoring, IP clearance, ...

Overview of the Project Creation Process



Project Activities

Timeline & future activities





The banner features a background image of a modern building with large glass windows and a thatched roof structure in the foreground. The text is overlaid on this image.

 **eclipsecon Europe**
Ludwigsburg, Germany, 3 - 5 November 2015

Evaluate the sessions at www.eclipsecon.org

+1 0 -1