HEADS
Heterogenous and Distributed
Services for the Future Computing Continuum

Why ThingML?
Installation/Execution

CPS-L0

CPS-L0: Why ThingML? Installation and Execution

- Why ThingML?
  - CPS with constrained resources
  - Abstraction
  - Separation of Concerns
- What is ThingML?
  - a domain-specific language
  - textual with visualization to UML
  - focused on asynchronous messaging between state machines

Why ThingML?

- The idea of ThingML is to develop
  - a practical model-driven software engineering tool-chain
  - which targets resource constrained embedded systems
    - such as low-power sensor and microcontroller based devices
- ThingML – thing modeling language
  - refers to Internet of Things – another way to say cyber-physical
- ThingML provides several compilers
  - for C, C++, Java, UML, etc
  - which makes its products usable and portable

ThingML and abstraction

- We want functionality, but we also want to get our (mental) arms around the problem
  - because we are going to improve our solution
  - because we are going to correct our solution
  - because we are going to evolve our solution
  - because we are going to enhance our solution
- We want constructs that help comprehend our system
- We want constructs that help distribute the development of the system

ThingML main constructs

- a **thing** is something that behaves as a state machine,
  - has local attributes
  - communicates asynchronously through ports
- **message**s are sent asynchronously between things
- **instance**s are established and connected
- ThingML contains a native action language

ThingML and separation of concerns

- Separation of concerns
  - often overlap with abstractions
  - are ways that allow us to focus on a manageable part of reality
  - are ways that makes it possible to distribute work
  - are ways that divides the work and makes it possible to plan
- We will go through many ways to separate concerns
  - PSM and PIM
  - Composite states
  - Concurrent things

The most updated installation instructions

- Follow the Readme.rd at:
  - https://github.com/SINTEF-9012/ThingML/blob/master/README.md
- Go to the tutorial of ThingML and walk through it: https://github.com/HEADS-project/training/tree/master/1.ThingML_Basics
  - Copy the “Hello World” model and execute it!

ThingML installation for Java starting from HEADS IDE

- ThingML is the modeling language for code generation
  - Look at http://thingml.org and http://heads-project.eu/
- Download HEADS-IDE from http://headside.gforge.inria.fr/
  - It includes Eclipse and most of the necessary plugins
- Install Plantuml from http://plantuml.sourceforge.net/updatesitejuno
- and Graphviz from www.graphviz.org
- Install a proper Oracle Java JDK (we do not use a JRE).
  - Configure the Eclipse to point to that Java JDK
    - Windows / Preferences / Java / Installed JREs
- Go to the tutorial of ThingML and walk through it: https://github.com/HEADS-project/training/tree/master/1.ThingML_Basics

Java and Maven

- We are going (in the first place) to use Java as our target language. In the sequel we may rather use C
- When using Java with ThingML, we also use Maven
  - What you have downloaded with the HEADS IDE also includes a Maven plugin
  - See also https://maven.apache.org/what-is-maven.html
- Note: you have to set a real JDK as the JRE
  - Window/Preferences/ click Java/InstalledJREs and then Add... and put in the jdk top folder.

How to compile and run

- Right-click on the configuration file
  - HEADS/ThingML
    - java/swing
  - There will be warning and notices, but beware of errors
- Open folder thingml-gen\java\CPS
  - Right-click on pom.xml
  - Run As
    - Maven Build
      - First time for a project a dialog comes up
      - Give it an appropriate specific name
      - Fill in Goal: *clean install exec:java*

Consortium


SINTEF


informatics mathematics
Inria


M2M2on


tell u


software AG


ATC
ATHENS TECHNOLOGY CENTER


HEADS
http://heads-project.eu


SEVENTH FRAMEWORK PROGRAMME