Model Transformations for Embedded System Design and Virtual Platforms

Nikos Matragkas, Ian Gray, Richard Paige, Dimitris Kolovos, Neil Audsley, Leandro Indrusiak

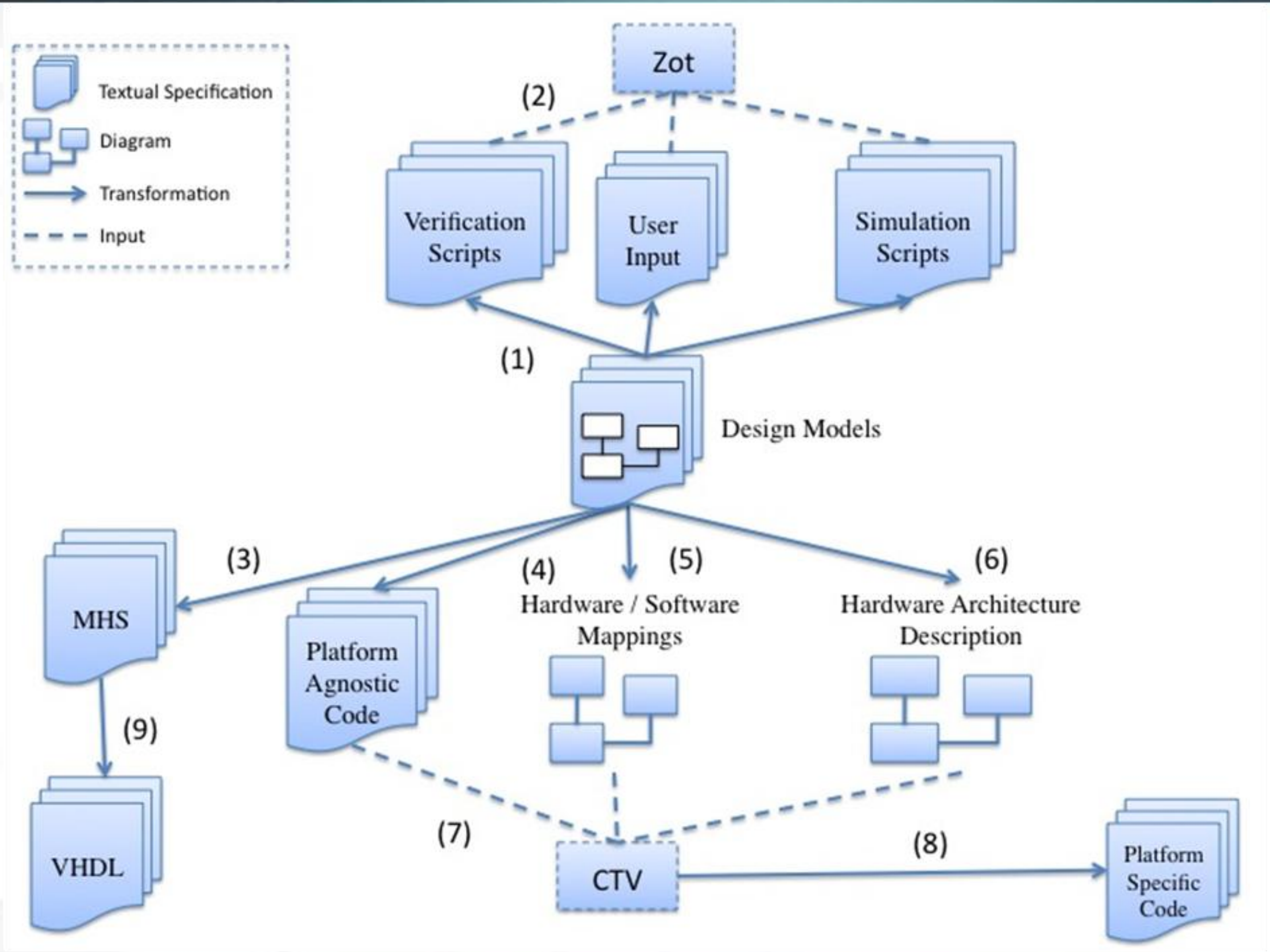
Department of Computer Science

The University of York

MDE for Embedded Systems

- MADES investigates a model-centric approach to embedded systems development
  - Models are the main artefacts of the development process
  - Models are automatically analysed, verified and then transformed into concrete artefacts
    - Hardware specifications, code, configuration files
- Aims: correctness, consistency, productivity, cost-effectiveness

MADES Artefacts and Workflow



MADES Design Models

- Specified in the MADES modelling language
  - Reuses parts of UML, MARTE and SysML
  - Tailored for Embedded Systems
  - 10 diagrams types
    - Requirements, High-Level Structure, Hardware
    - Software, Allocation, Time
    - Activity, Sequence, State, Use case, Interaction (ext)
- Tool support
  - Modelio (Softeam), Papyrus (open-source)

Model Transformation in MADES

- **Aim:** Transform MADES design models into other representations automatically
  - To achieve consistency, productivity and correctness by construction
- Two types of model transformations
  - Model-to-model (M2M) transformations
  - Model-to-text (M2T) transformations

Enabling Technologies: EMF

- Eclipse Modelling Framework (EMF)
  - Sub-project of Eclipse Modelling
  - Supports the definition of modelling languages
  - Standards compliant XMI model serialisation
    - Modelio exports models in XMI, Papyrus built on EMF
  - Mature open-source project
  - Lots of tooling built atop it
    - e.g. GMF, Graphiti for graphical editors

Enabling Technologies: Epsilon

- Epsilon (www.eclipse.org/gmt/epslon)
  - Sub-project of Eclipse Modeling
  - Provides consistent and interoperable languages for model transformation
    - M2M and M2T transformation
    - + languages for model validation, comparison, refactoring
  - Seamless integration with Eclipse/EMF
    - Eclipse-based editors, launchers, ANT tasks

Enabling Technologies: M2T

- Epsilon Generation Language (EGL)
  - Template-based (i.e. like PHP/JSP)
  - Metamodel-agnostic
    - Can generate text/code from any EMF-based model
  - Target-agnostic
    - Can generate text/code in any language
  - Support for preserving hand-written code
    - Protected regions in templates

Enabling Technologies: M2M

- Epsilon Transformation Language (ETL)
  - Rule-based
    - Automated rule scheduling
  - Hybrid style
    - Imperative style can be used for complex transformations
  - Can access an arbitrary number of models
    - Not only suitable for 1-1 transformations
  - Interoperates seamlessly with EGL (M2T)

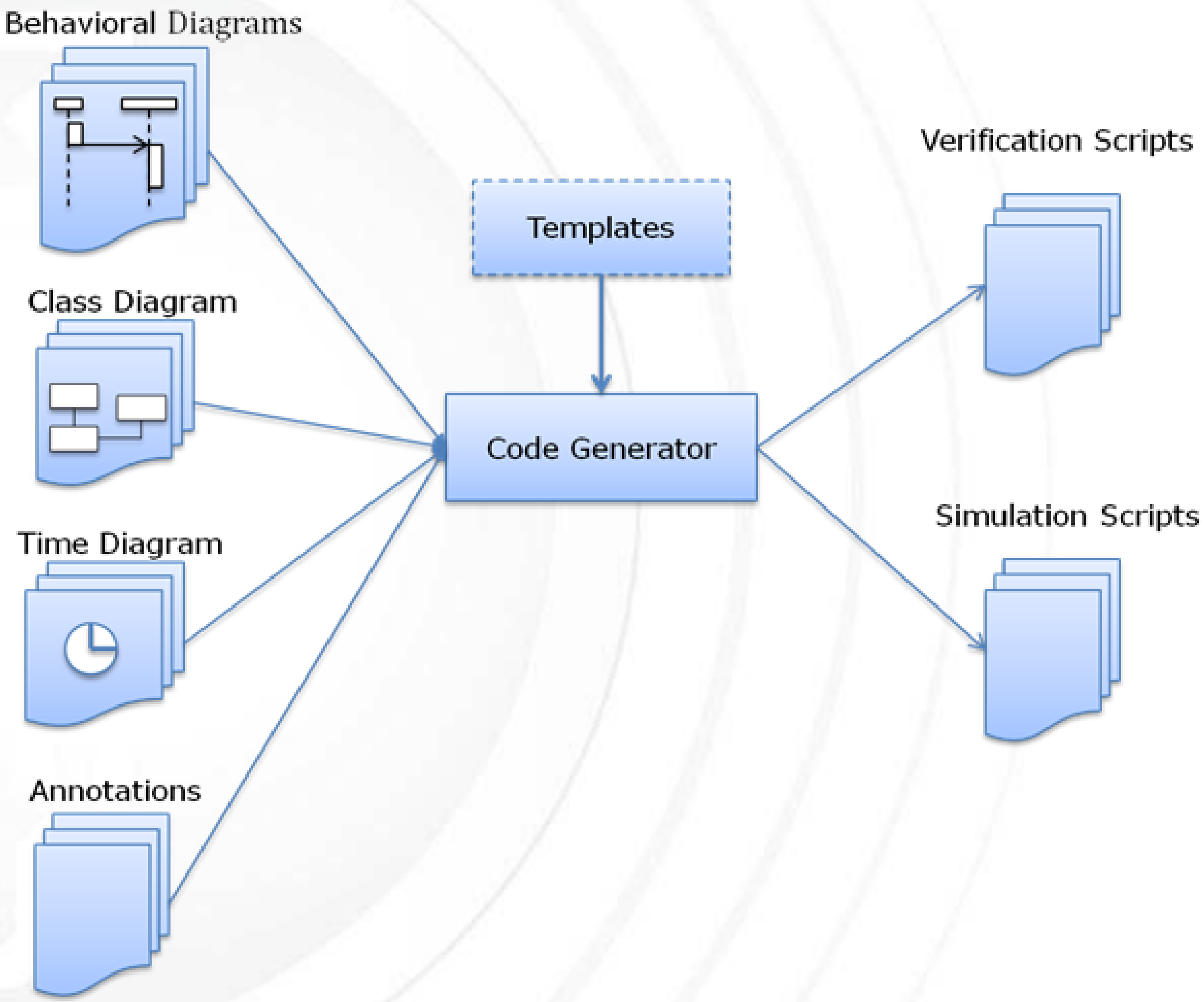
Enabling Technologies: M2T

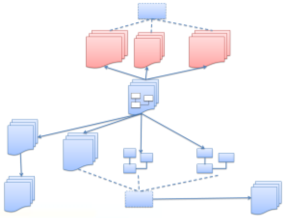
- EMFText (http://www.emftext.org)
- Specify textual syntaxes for (modelling) languages
  - Parser generator on steroids
  - Produces, parser, fully-blown editor, M2T, T2M from a grammar

emftext
concrete syntax mapper

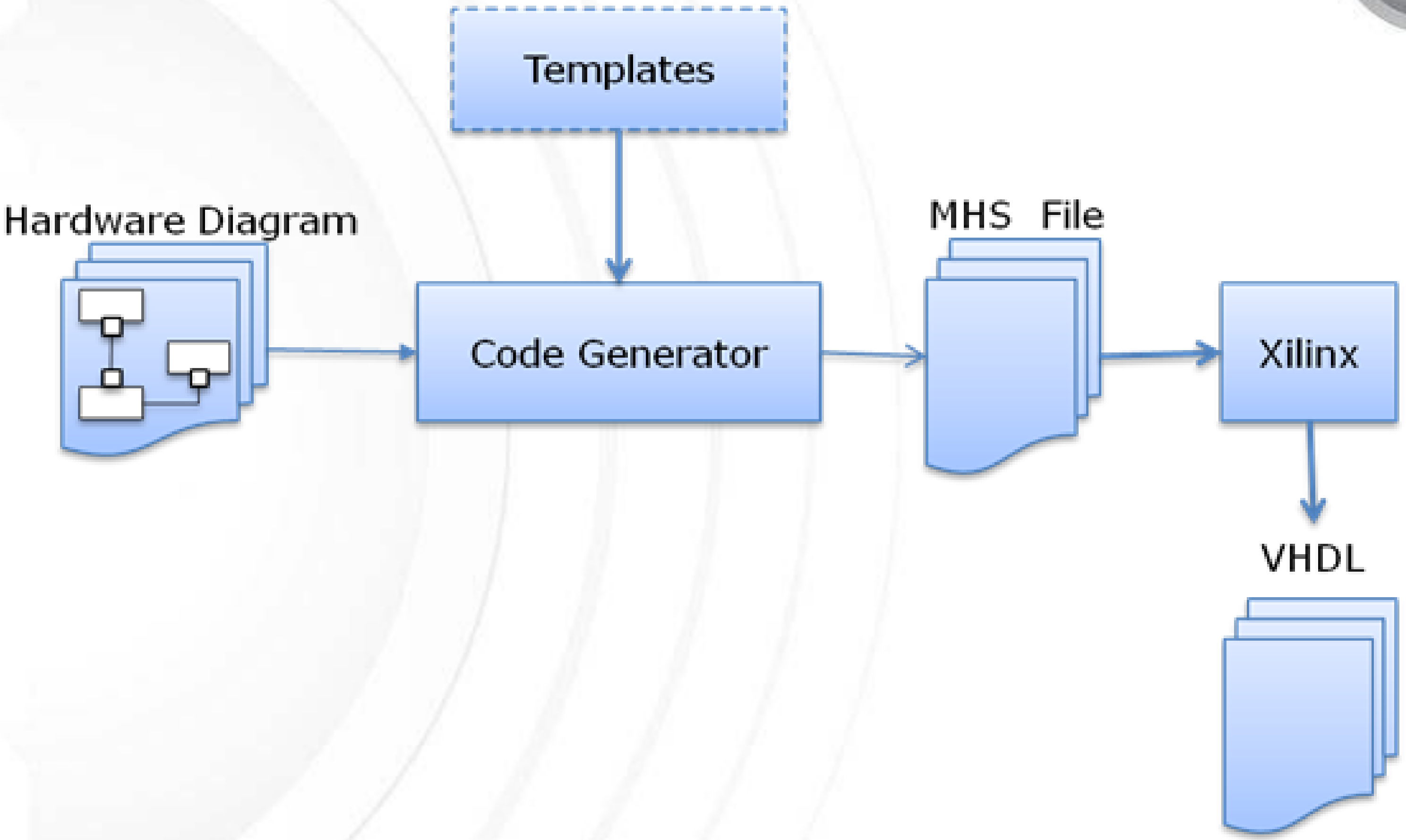
- Use EMFText to develop support for MHS
  - Existing language for Microprocessor Hardware Specification

MADES to Zot M2T Transformation



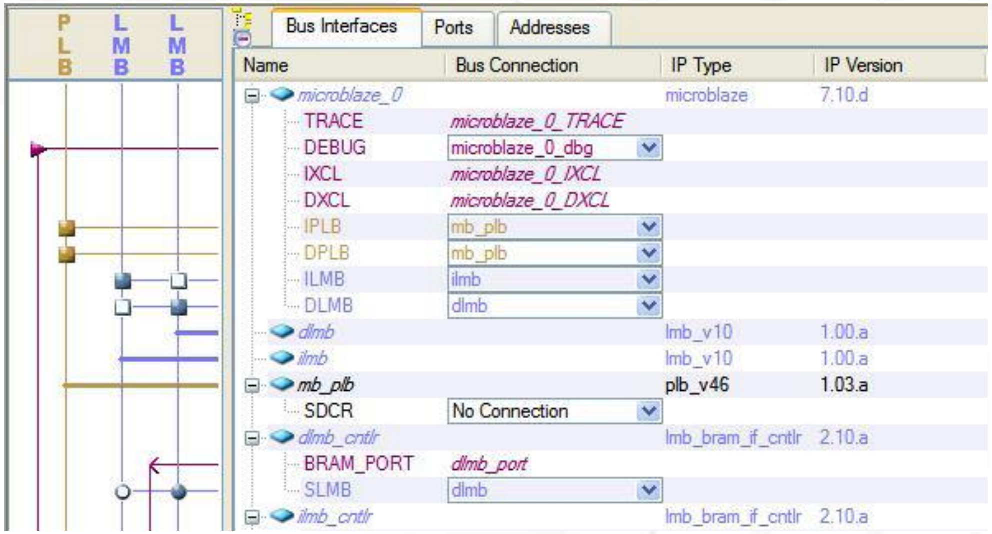


MADES to Hardware Specification

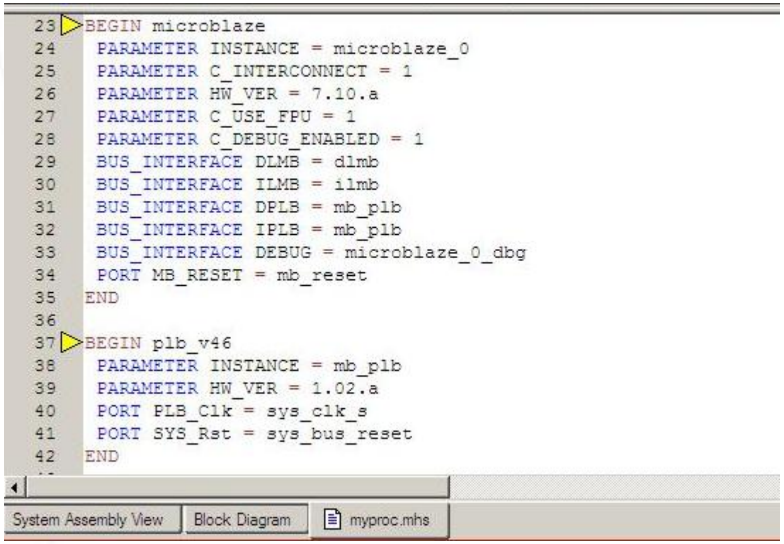


*MHS: Microprocessor Hardware Specification

MADES to Hardware Specification



Name	Bus Connection	IP Type	IP Version
microblaze_0		microblaze	7.10.d
TRACE	microblaze_0_TRACE		
DEBUG	microblaze_0_dbg		
IXCL	microblaze_0_IXCL		
DXCL	microblaze_0_DXCL		
IPLB	mb_plb		
DPLB	mb_plb		
ILMB	ilmb		
DLMB	dlmb		
dlmb		lmb_v10	1.00.a
ilmb		lmb_v10	1.00.a
mb_plb		plb_v46	1.03.a
SDCR	No Connection		
dlmb_cntlr		lmb_bram_if_cntlr	2.10.a
BRAM_PORT	dlmb_port		
SLMB	dlmb		
ilmb_cntlr		lmb_bram_if_cntlr	2.10.a

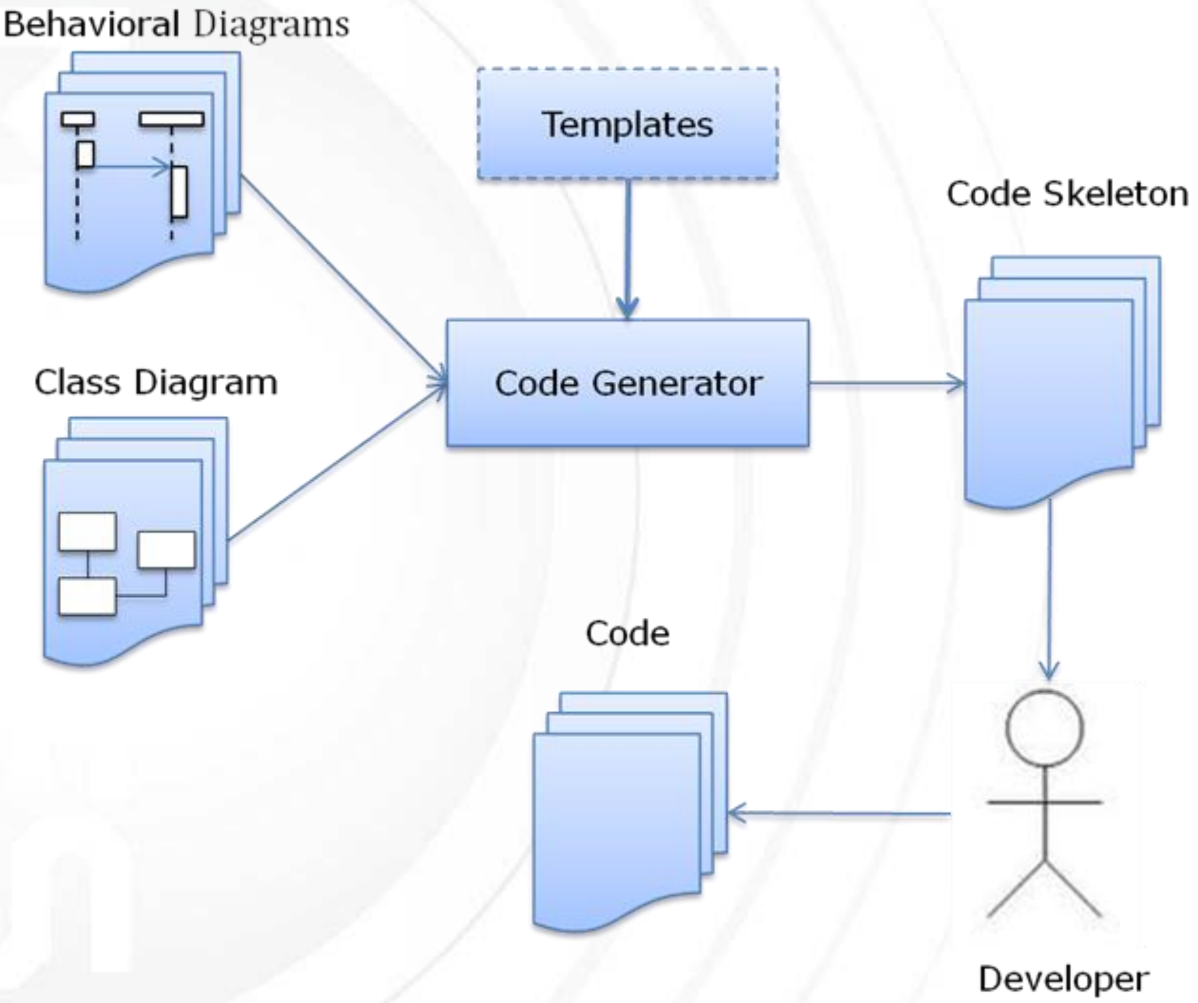


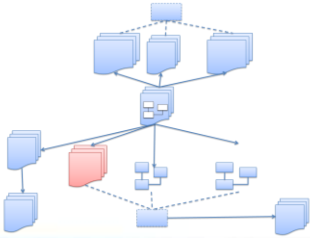
```
BEGIN microblaze
 PARAMETER INSTANCE = microblaze_0
 PARAMETER C_INTERCONNECT = 1
 PARAMETER HW_VER = 7.10.a
 PARAMETER C_USE_FPU = 1
 PARAMETER C_DEBUG_ENABLED = 1
 BUS_INTERFACE DLMB = dlmb
 BUS_INTERFACE ILMB = ilmb
 BUS_INTERFACE DPLB = mb_plb
 BUS_INTERFACE IPLB = mb_plb
 BUS_INTERFACE DEBUG = microblaze_0_dbg
 PORT MB_RESET = mb_reset
END

BEGIN plb_v46
 PARAMETER INSTANCE = mb_plb
 PARAMETER HW_VER = 1.02.a
 PORT PLB_Clk = sys_clk_s
 PORT SYS_Rst = sys_bus_reset
END
```

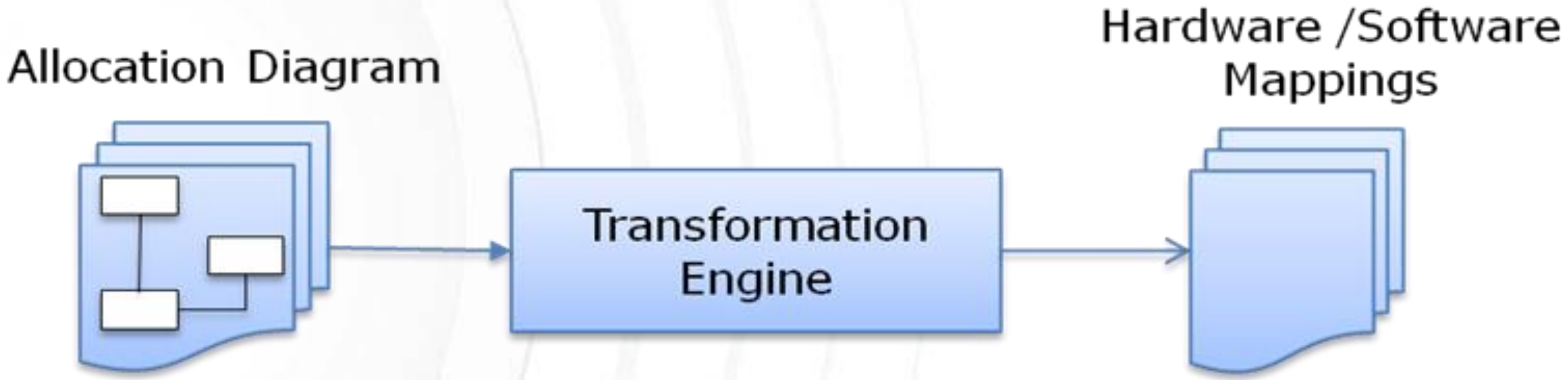
The MHS (Microprocessor Hardware Specification) file is the main source file representing the hardware part of the embedded system. This file contains the processor and all peripheral instantiations along with their parameters.

MADES to Platform-Agnostic Code





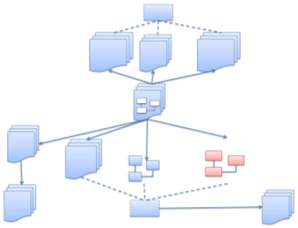
Model Transformations in MADES



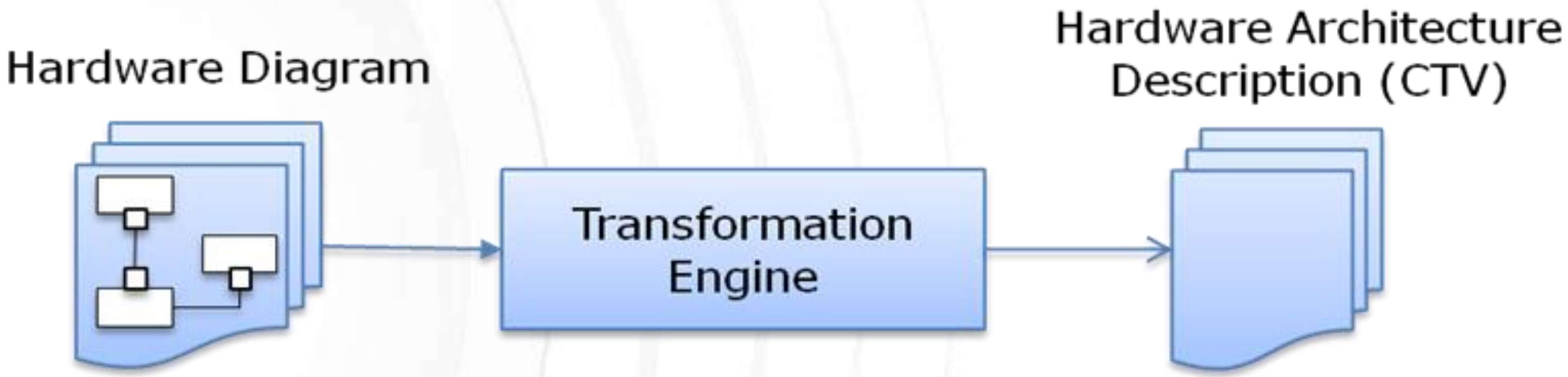
Allocation Diagram

Transformation Engine

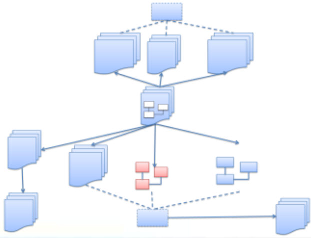
Hardware /Software Mappings



MADES to Hardware Architecture for CTV



*CTV: Compile-Time Virtualisation



Conclusions

- Model Transformation is an essential part of the MADES methodology
  - Enhances automation, consistency
  - Facilitates interoperation between different tools of the tool-chain
    - MADES -> Zot, MADES->MHS
  - Transformations help to evaluate the expressiveness of the MADES language
  - Transformations specify the semantics of the MADES language