DIPLODOCUS: An Environment for the Hardware/Software Partitioning of Complex Embedded Systems

Ludovic Apvrille, ludovic.apvrille@telecom-paristech.fr

ETR’2013, Toulouse, France
Introduction
The DIPLODOCUS Approach
Outlook

Goals

▶ To share an experience of real-time systems modeling
▶ To propose a language, a tool, and a method dedicated to the partitioning of complex and real-time embedded systems
  ▶ DIPLODOCUS, a modeling language based on SysML
  ▶ TTool for model simulation and user-friendly formal verification
  ▶ A method that applies to a broad variety of real-time systems
▶ To answer your questions
Introduction

Context: Model-Driven Engineering
Our contribution
Design Space Exploration

The DIPLODOCUS Approach
DIPLODOCUS in a Nutshell
Methodology

Outlook
Outline

Introduction

Context: Model-Driven Engineering
Our contribution
Design Space Exploration

The DIPLODOCUS Approach

Outlook
Designing Embedded Systems

How to Handle Complexity?
Modeling and verification!
(But there are other options)
Modeling is not Really a New Technique. . .

. . . and it is not limited to Software!
Abstraction Level

How to create a stable data model

(source: peek and Poke, July, 2013)
Software Development Techniques for E. S.

**Code-based approaches**
- Extreme Programming
  - Strongly tested step-by-step code increments
- Agile Software Development
  - Focus on change in specification

**Model-based approaches**
- V-Cycle
  - KAOS, AADL, MDE, ...
- Formal models
  - B, LOTOS, Petri nets, ...

- Code-based approaches

- Model-based approaches
Model Driven Engineering

Definition

- Process based on abstract graphical representations for a given domain
- Intends to improve software engineering quality criteria
  - Reliability, extensibility, maintainability, ...
- Should enhance team communication and documentation

Abstraction levels

- Platform Independent Model, Platform Specific Model
- Model transformations
UML Profiles

Definition

- UML defines extension mechanisms to e.g.,
  - Define new operators
  - Provide a semantics
  - Give a methodology

Example of profiles

- Profiles defined by OMG (e.g., SPT, MARTE, SysML)
- Profiles defined by tool vendors (e.g. in Rhapsody, Artisan)
- User-defined and company-defined models
UML Profiles and MDE

*UML profiles are a way to define domain-specific languages for MDE*

**Our contribution in MDE**

Definition of UML profiles for modeling and verifying complex embedded systems

Definition of methodologies based on the V-cycle

Definition of model transformations for simulation, formal verification and code generation purpose

Implementation in a toolkit (TTool)
TTool: A Multi Profile Platform

**TTool**
- Open-source toolkit mainly developed by Telecom ParisTech / COMELEC
- Multi-profile toolkit
  - DIPLODOCUS, AVATAR, ...
- Support from academic (e.g. INRIA, ISAE) and industrial partners (e.g., Freescale)

**Main ideas**
- Lightweight, easy-to-use toolkit
- Simulation with model animation
- Formal proof at the push of a button
A System-On-Chip = set of SW and HW components intended to perform a predefined set of functions for a given market

- Constraints
  - Right market window
  - Performance and costs
Introduction
The DIPLODOCUS Approach
Outlook

Context: Model-Driven Engineering
Our contribution
Design Space Exploration

Design Challenges

Complexity
- Very high software complexity
- Very high hardware complexity

Problem
How to decide whether a function should be implemented in SW or in HW, or both?

Solution
Design Space Exploration!
Design Space Exploration

- Analyzing various functionally equivalent implementation alternatives
- → Find an optimal solution

Important key design parameters

- Speed
- Power Consumption
- Silicon area
- Generation of heat
- Development effort
Level of Abstraction

Problematic

- Designers struggle with the complexity of today’s circuits
- Cost of late re-engineering
  - Right decisions should be taken as soon as possible ...
  - And quickly (time to market issue), and so, simulations must be fast

→ System Level Design Space Exploration

Reusable models, fast simulations / formal analysis, prototyping can start without all functions to be implemented

But: high-level models must be closely defined so as to take the right decisions
Outline

Introduction

The DIPLODOCUS Approach

DIPLODOCUS in a Nutshell

Methodology

Outlook
DIPLODOCUS in a Nutshell

DIPLODOCUS = UML Profile

- System-level Design Space Exploration
- Y-Methodology
- MARTE compliant

Main features

- Data are abstracted
- Formal semantics
- Very fast simulation support
- Fully supported by an open-source toolkit
  - TTool
Partitioning with the Y-Methodology

Diagram:
- Partitioning
  - Simulation Formal verification
  - Application model
    - Simulation Formal verification
  - Architecture model
    - Mapping model

Methodology:
Partitioning with the Y-Methodology

Introducing the DIPLODOCUS Approach

Outlook

DIPLODOCUS in a Nutshell
Application Modeling

Functions are first modeled independently from the architecture.

Simulation
Static analysis

Application modeling

Architecture modeling

Mapping

DSE

Simulation
Static analysis
Architecture Modeling

Simulation
Static analysis

Application modeling

Architecture modeling

Simulation
Static analysis

DSE

Then, architecture is modeled based on generic hardware components: microprocessors, buses, memories, bridges, etc.
Mapping

Application modeling

Architecture modeling

Simulation
Static analysis

DSE

Functions are then associated to architecture components

Static analysis

mapping
Browsing the DIPLODOCUS Methodology

- Application structure
- Application behavior
- Formal verification
- Architecture model
- Mapping model
- Simulation
- Formal verification

Application modeling

Architecture modeling

DSE

Simulation
Static analysis

Mapping

Simulation
Static analysis
Application Structure (Smart Card system)
Application Behavior

Activity Diagram of the SmartCard component

The first two layers of the OSI model are used to communicate between the smart card and the terminal.
Formal Verification at Application Level

- No assumption on the underlying architecture
- All possible interleavings between actions are considered
- Formal verification is based on LOTOS/CADP or UPPAAL
  - Press-button approach
Architecture

- Given in terms of parameterized nodes
- CPU, HWA, Bus, Memory, Bridge, etc.
- CPU parameters: scheduling policy, cache miss ratio, miss-branching prediction, pipeline size, etc.
Task are mapped on execution nodes (e.g., CPUs, HWAs)

Channels are mapped on communication and storage nodes
After-Mapping Simulation

- **TTool Built-in simulator**
- **Extremely fast**
- **Diagram animation**
- **Step-by-step execution, breakpoints, etc.**
After-Mapping Simulation (Cont.)

- **CPU0**
  - AppC::TCPIP: 50%, 195 pW
  - AppC::SmartCard
  - AppC::Application

- **CPU1**
  - AppC::InterfaceDevice: 66%, 230 pW

- **CPU2**
  - AppC::Timer: 0%, 90 pW

- **Bus0**
  - Memory0: 0%

- **After-Mapping Simulation (Cont.)**

Back to methodology
After-Mapping Formal Verification

- TTool built-in simulator can compute all possible execution paths
- Graph analysis and visualization
After-Mapping Coverage-Enhanced Simulation

Possibility to select a given part of the model to be explored

- Minimum percentage of operators coverage
- Minimum percentage of branch coverage

Implementation: TTool built-in model-checking techniques
Results

Fully integrated environment for the partitioning of Systems-on-Chip and complex embedded systems

- Based on UML
- Open-source toolkit

- Partners: Texas Instruments, Freescale, European project EVITA, European project SACRA, LIP6, ...
- 2 Ph.D. completed (Chafic Jaber, Daniel Knorreck, Jair Gonzalez-Pina), 2 on-going Ph.D. (Fériel Ben Abdallah, Andrea Enrici)
A Few Case Studies ...

- MPEG coders and decoders (Texas Instruments)
- LTE SoC (Freescale)
- Partitioning in vehicle embedded systems (EVITA project)
- Partitioning and code generation for Software-Defined Radio systems (SACRA project)
To Go Further ...

TTool and DIPLODOCUS

ttool.telecom-paristech.fr