Séminaire Supélec/SCEEE

Models driven co-design methodology for SDR systems

LECOMTE Stéphane
Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA co-design methodology: MOPCOM
- MDA tools
- Experiments
- Conclusion
Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA Co-design Methodology : MOPCOM
- MDA tools
- Experiments
- Conclusion
Challenge

- Design of real time embedded systems
  - More and more complex systems
  - Heterogeneous systems

- Technology of digital chip improving quickly
  - Integrating a system into one chip
    - SoC : System on Chip => ASIC*
    - SoPC : System on Programmable Component => FPGA**

- Shorter and shorter Time-to-Market

---

* ASIC : Application-Specific Integrated Circuit
** FPGA : Field Programmable Gate Array
State of the art

Today the co-design methodologies do not progress as quickly as the technology

- Rupture of design process
- Different process
  - For hardware
  - For embedded software
- Specific tools
  - For hardware design (EDA tools)
  - For embedded software
- Integration and Validation
  - Too long
  - Too many difficulties
## Solutions

<table>
<thead>
<tr>
<th>Problems</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increasing complexity, Decreasing Time-to-Mark</td>
<td>➢ High level approach to increase productivity</td>
</tr>
<tr>
<td></td>
<td>➢ Portability, functionality/architecture</td>
</tr>
<tr>
<td></td>
<td>independence</td>
</tr>
<tr>
<td></td>
<td>➢ Component-based approach</td>
</tr>
<tr>
<td></td>
<td>➢ Reuse</td>
</tr>
<tr>
<td>Communications between teams</td>
<td>➢ Common formalism for system/software and</td>
</tr>
<tr>
<td></td>
<td>hardware engineer</td>
</tr>
<tr>
<td>Obsolescence</td>
<td>➢ Capitalize knowledge and experience</td>
</tr>
<tr>
<td>Quality of process</td>
<td>➢ Process formalization</td>
</tr>
<tr>
<td></td>
<td>➢ Traceability and test improvement</td>
</tr>
</tbody>
</table>

Use the same design process and tools for hardware and embedded software development
Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA Co-design Methodology: MOPCOM
- MDA tools
- Experiments
- Conclusion
Objectives of thesis

- **Formalization of a new development process based on high level models for Co-design for SoC/SoPC**
  - Covers Electronic System Level (ESL) domain
  - Use [UML](https://www.uml.org) models
  - Use [MARTE](http://www.omg.org) profile from OMG*, extension of UML
  - Use [Model Driven Architecture](http://www.omg.org) (MDA) approach
  - Automatic [code generation](http://www.omg.org)
  - Generation of documentation

- **Integration of technology of partial dynamic reconfiguration of FPGA (reconfigurable hardware for SoPC)**

Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA Co-design Methodology: MOPCOM
- MDA tools
- Experiments
- Conclusion
Model Driven Architecture (MDA)

- Based on model transformations to formalize and to automate the design process

MDA Process based on several model types
- Platform Independent Model (PIM)
- Platform Model (PM)
- Platform Specific Model (PSM)

Use the modeling language: Unified Modeling Language
- Standardized language by the OMG
- Graphical & annotated language for modeling high level design approach
- UML describes structural and behaviour aspects of the systems
Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA Co-design Methodology : MOPCOM
- MDA tools
- Experiments
- Conclusion
MOPCOM co-design Methodology

Modélisation et spécialisation de Plates-formes et Composants MDA
A Design Process based on UML Profiles

- UML is a unified language but not a methodology
  - How to design hardware with UML?

- Integration with system and software processes

- UML extension to RTE systems
  - MARTE (Modeling and Analysis of Real Time and Embedded systems) Profile
    - Modeling time constraints
    - Modeling Hardware
    - Modeling Allocation
    - Performances analysis
  - Huge set of concepts
  - No methodology to support activity based on MARTE
MOPCOM Abstraction levels (1/3)

Abstract Modeling Level (AML)

- Modeling of high level of abstraction
- Validation of functional architecture and behavior
MOPCOM Abstraction levels (2/3)

Execution Modeling Level (EML)

- Modeling the topology of hardware platform
- Add information of time constraints
- Dedicated to architecture exploration

Diagram:

- Requirements Analysis
  - Definition of system use cases
  - Functional application (PIM)
  - Abstract Platform, MoC component (PM)
  - Allocated Model (PSM)

- MoC Analysis (SystemC model)
- Topology & Schedulability Analysis (SystemC model)
- Execution Modeling Level
  - AML (PIM)
  - Execution Platform (PM)
  - Allocated Model (PSM)

09/24/2009
MOPCOM Abstraction levels (3/3)

- Detailed Modeling Level (DML)
  - Detailed modeling hardware platform
  - Enable to HLS tools (C/C++ code generation)
  - Enable to VHDL code generation

- Requirements Analysis
  - Definition of system use cases

- Functional application (PIM)
- Abstract Platform, McC component (PM)
- Allocated Model (PSM)
- MoC Analysis (SystemC model)

- Abstract Modeling Level

- AML (PIM)
- Execution Platform (PM)
- Topology & Schedulability Analysis (SystemC model)
- Allocated Model (PSM)

- Execution Modeling Level

- AML (PIM)
- Detailed Platform (PM)
- C/C++ for HLS tools
- Allocated Model (PSM)

- Detailed Modeling Level

- Software
- Soft Driver
- Hard Driver
- RTL Code (VHDL)
MOPCOM flow

- Three levels of modeling
  - Each level use MDA approach
  - Modeling with UML and MARTE

- Formalization of process
  - A meta-model describes the process
  - Associated modeling constraints for each level

- MOPCOM Profile
  - Add concepts that do not exist in UML and MARTE

- Iterative design process
Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA Co-design Methodology: MOPCOM
- Tools environment in MOPCOM
- Experiments
- Conclusion
MDA tools

Process (methodology) → Kermeta (metamodeling) → UML/MARTE Metamodel → Papyrus (modeling) → Java/EMF (transformation & generation) → Generated code

Methodological Rules (architecture, functional, allocation)

Open Source Capitalization

User entry: system specification → Tools instanciation → Scripts
MOPCOM tools

Process (methodology)

Kermeta (metamodeling)

Methodological Rules (architecture, functional, allocation)

UML/MARTE Metamodel

Open Source Capitalization

Rhapsody (modeling)

MDWorkbench (transformation & generation)

User entry: system specification

MOPCOM Tools instanciation

MDWorkbench scripts

Generated code (RTL, C, C++)
Code generator integrated in Rhapsody

Seamless integration in Rhapsody-in-C++ 7.5

- VHDL Configuration
- DML, Application & Platform Packages
- Hardware Libraries & Types
- MARTE & MOPCOM Profiles
- External Generator based on RulesComposer & RulesPlayer
- OMD for Application & Platform
- Generation from Statecharts
- Definition of VHDL properties
- Edition of VHDL code
- Logs & Build Links with EDA tools

09/24/2009
Outline

- Context
- Objectives of thesis
- Definitions/Vocabulary
- MDA Co-design Methodology: MOPCOM
- Tools environment in MOPCOM
- Experiments
- Conclusion
Test applications

Goal

- Validation of MOPCOM co-design methodology
- Validation the MDA tools instance in MOPCOM

Evaluation

- Comparison with traditional co-design flow
  - Profits (time and cost)
- Portability of MOPCOM methodology in others context
- Reusability of process with others MDA tools
Supelec experiment for MOPCOM

Limited SDR system

Constellation

16-QAM

QPSK

roll-off=0.22

roll-off=0.015

09/24/2009
DML Model

Platform Model

Identify this PLD resource to a reconfigurable resource with a specific tag

Used to manage the partial reconfiguration

09/24/2009
DML Model

Allocation

09/24/2009
THOMSON
TECHNICOLOR
Supélec
SystemC model

- Programmer’s View (untimed)
  - Equivalence with AML

<table>
<thead>
<tr>
<th>Modeling level</th>
<th>SystemC</th>
<th>OSCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Untimed Functional</td>
<td>Programmers View</td>
<td></td>
</tr>
<tr>
<td>Timed Functional</td>
<td>Transaction Level Modeling</td>
<td>PV + Timing</td>
</tr>
<tr>
<td>Cycle Accurate Bus Accurate</td>
<td>Cycle Callable</td>
<td></td>
</tr>
<tr>
<td>Register Transfer Level</td>
<td>RTL SystemC</td>
<td>RTL</td>
</tr>
</tbody>
</table>

- Equivalence with AML

09/24/2009
Outline

- Context
- Objectives of thesis
- Definitions
- MDA Co-design Methodology : MOPCOM
- Tools environment in MOPCOM
- Experiments
- Conclusion
Conclusion

Feedback

- Portability of methodology is difficult
  - UML tools makes specific/proprietary model interpretation
  - Reuse of models is difficult
- Existing code generators are not complete
- First co-design methodology using MARTE profile for modeling RTE system
- Same process for design hardware and software

Future works

- Code generator fully integrated inside the modeling tool
- Updating the code generators
- Use the methodology in others domains
Acknowledgement

- Partners of MoPCoM SoC/SoPC project*
  - Thalès (Airborne Systems)
  - Thomson (Corporate Research)
  - Sodius
  - ENSIETA
  - Lab-STICCC (UBS)
  - INIRIA (Triskell team)
  - Supelec (SCEE team)

MOPCOM web site : www.mopcom.fr
Thanks!

Questions and Discussions