JEAN-CHRISTOPHE LE LANN
ENSEIGNANT-CHERCHEUR

EMBEDDED SYSTEMS, MODEL-DRIVEN ENGINEERING
From System-level models to heterogeneous embedded systems

Jean-Christophe Le Lann
Joel Champeau
Papa Issa Diallo
ENSTA-Bretagne / Labsticc

Pierre-Laurent Lagalaye
Modaë Technologies
Overview

• Introduction
• Experimental toolchain
• Models of computation
• Result and discussion
• Conclusions
Introduction

• History
  – 1999: Thomson Multimedia / Technicolor
    • System-level specification for SoC design + Mopcom ANR
    • Video compression system
    • Multicore + SIMD ~15M gates
  – Needs:
    • ease of algorithm capture: data + high-level control flow
    • Simulation: untimed, functional, data movements, events
    • Synthesis/compilation on heterogeneous platforms
    • Allowing quick iterations in the design flow
  – 2009: startup Modaë Technologies
    • Interpreted languages as input + DSL
New needs

• **System-engineering** practical aspects
  – IBM Rhapsody + UML 2.0 at the front

• **Software engineering** for embedded systems
  – Not only algorithmic, nor event-driven
  – Importance of object-oriented

• Need for **openness**
  – insurance of independence wrt tool providers
  – Quite different from classical ESL business-model
  – Facilitate toolchains development
**Metamodeling**  
for tool development

**MDD**: model-driven development

---

**Business knowledge**

**Model edition**

**Code generation, ...**

---

Eclipse EMF support  
Sodius MDWorkbench, ...
Experimental toolchain

Modeling in UML 2.0
Transformation scripts in MDWorkbench
« Backend »
System-level synthesis
Modaë SLS
IBM Rhapsody UML 2.0

Diagramme de **composants** (éventuellement composite)

**Statecharts** associé dans le cas de **classes actives**
System modeling with Modaë

Ruby/Python algorithms, object-oriented

Addition of an internal DSL
...graphical

Network.new('example') do
  p1=MyProcessing.new('p1')
  p2=....
  ...
  connect :fifo_5, p1.o => p2.i
  ...
end

...textual
System modeling with Modaë
System modeling with Modaë
Behavioral blocks in Modaë

A block model is a class that our toolchain understands needs to be able to exchange data with other blocks. It has inputs/outputs, a constructor, and traditional Ruby instance variables which can be seen as variable states. You put your algorithm here, access to inputs/outputs via send/receive, and organise the internal processing into method calls.
Modaë SLS (2/2)

CDFG: Control-data flow graph

Synthesizable HDL code

Internal representation
Architecture template

- Multithreaded C code
- C driver / VHDL communication registers
- VHDL RTL algorithmic code

- RISC processor
- DMA
- External DDR

- FPGA/SoC

- Xilinx Spartan 6 FPGA
- Freescale IMX Bus
- ARM 9 running Linux
Preliminary results

Simple UML 2.0 model – simple action language

HW/SW mapping annotated

Software synthesis + HLS

VHDL RTL synthesis

Porting on platform
Conclusions

- Ceremonial system-level processes vs agile processes and languages
  - Possible interactions
  - Complementary
- MDD : several technologies to develop a system
  - Endogeneous vs Exogeneous battle ?
- Example :
  - Is UML easier / more expressive then Ruby +DSL ?
QUESTIONS?

Jean-christophe.le_lann@ensta-bretagne.fr