Model Driven Embedded Systems

Ian Oliver

Nokia Research Center

Helsinki, Finland

20 June 2003

Contents

- UML
- MDA
  - Top down
  - Embedded Concerns
  - Meta-modelling
  - Model Matrix
  - Model mappings
- DSP Development
- PUSSEE and proven design
- Future work

UML

- **Unified Modelling Language*(S)***

- Standard representations of modelling elements for object oriented concepts:

  - Class, object, state etc
  - Action Language (action semantics)
  - Constraint Language (Object constraint language – OCL)

- Weakest semantics

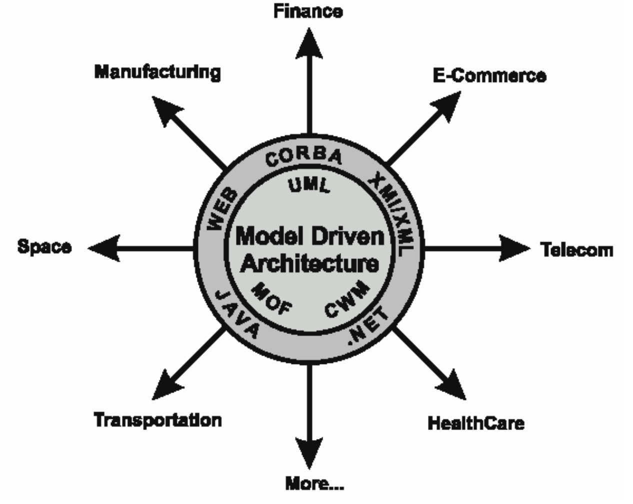
  - Eg: State diagrams, sequence diagrams (MSCs in the future)

- Stronger semantic support provided through profiling

  - UML-RT, SPEM, Copenhagen

MDA

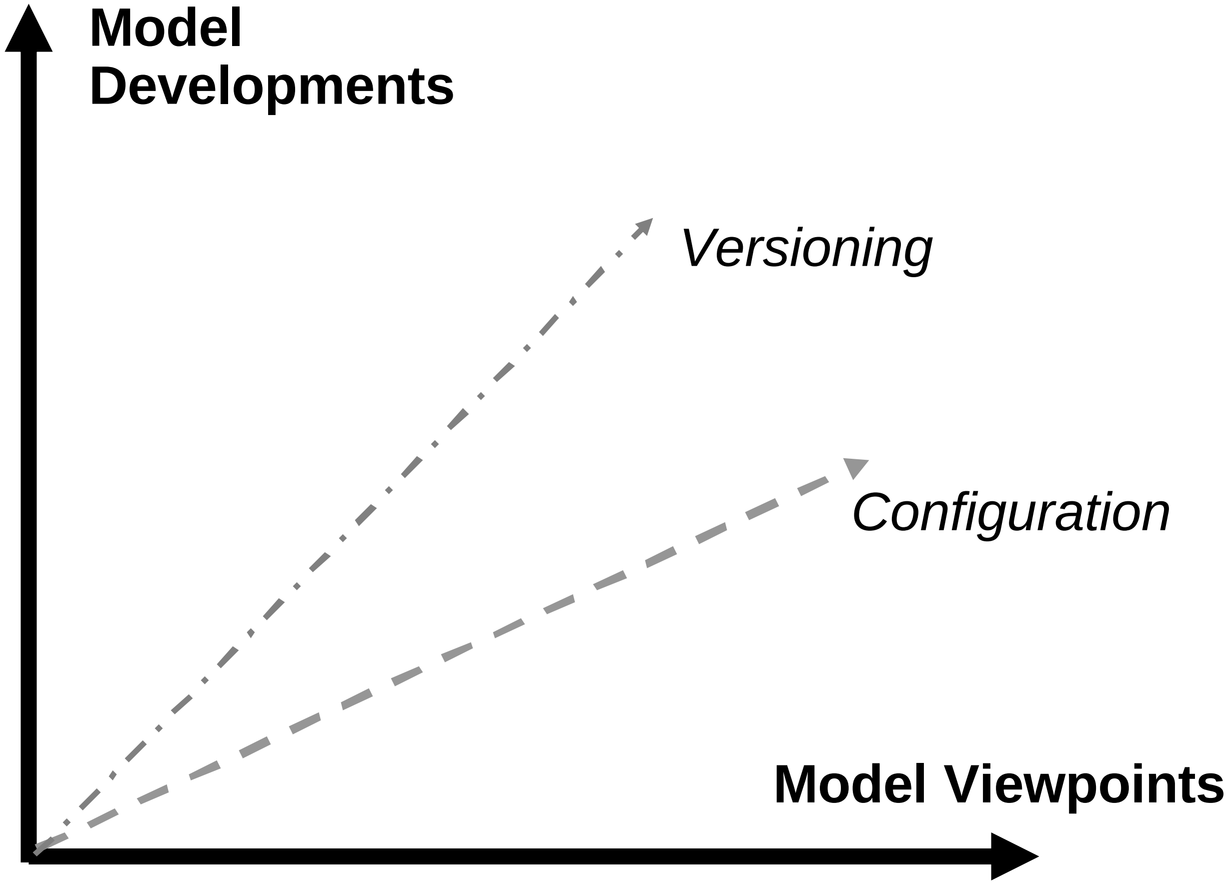
- **Model Driven Architecture**
  - OMG’s new, great, “solves everything”, idea.



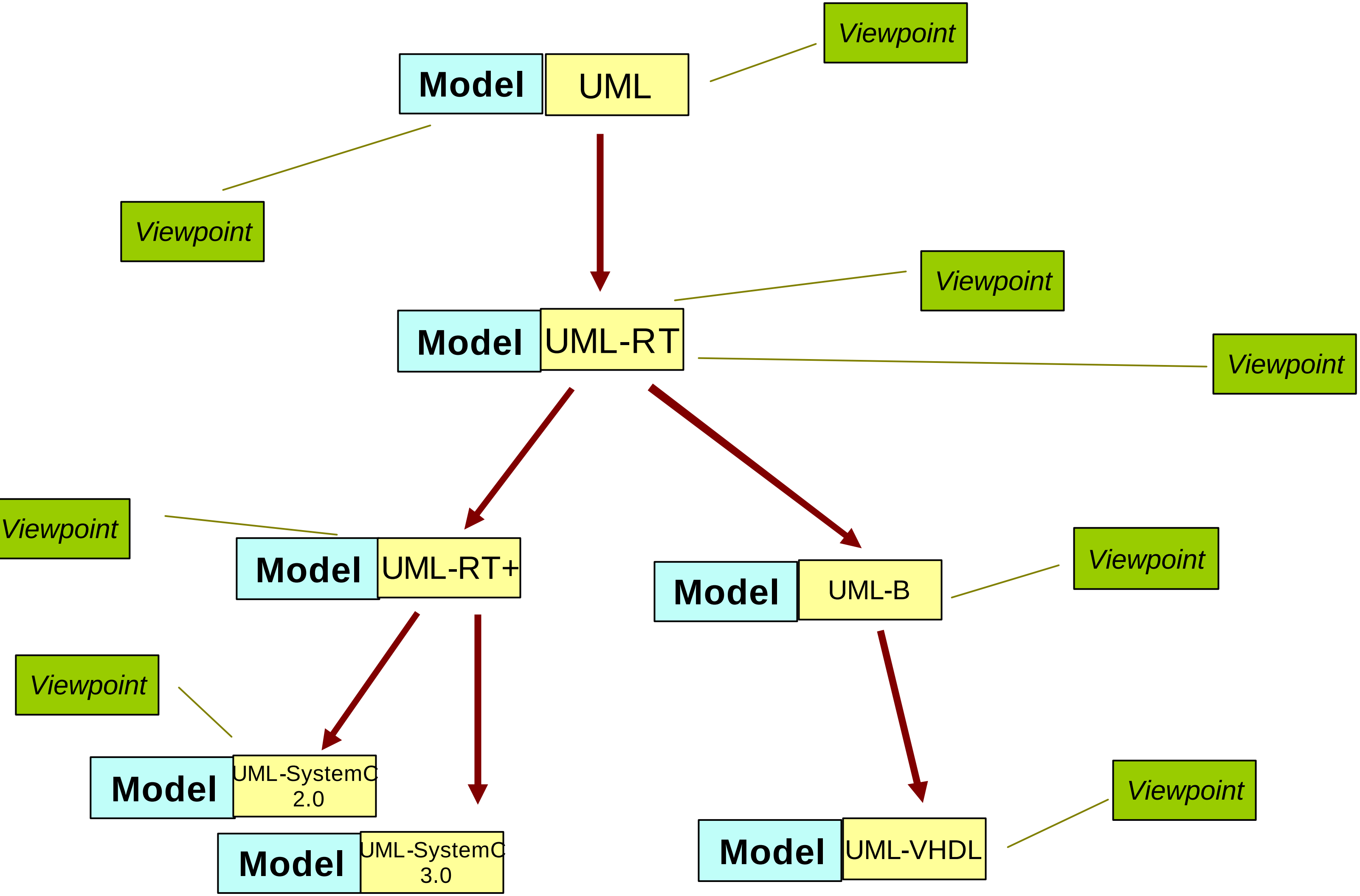
- **Basic idea:**
  - Platform Independent Models (PIMs)
    - *are mapped into*
  - Platform Specific Models (PSMs)

- Relies heavily upon meta-models describing the various domains in which one works
- Mappings between the PIM and PSM domains
- Assumed to support *(strictly?)* top-down design...

Model Matrix



MDA Example



Viewpoint

Model UML

Viewpoint

Viewpoint

Model UML-RT

Viewpoint

Viewpoint

Model UML-RT+

Model UML-B

Viewpoint

Viewpoint

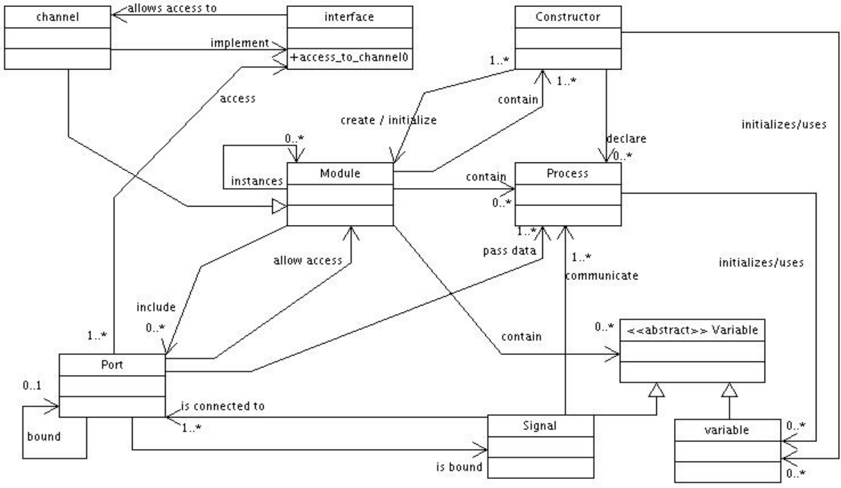
Model UML-SystemC 2.0

Model UML-SystemC 3.0

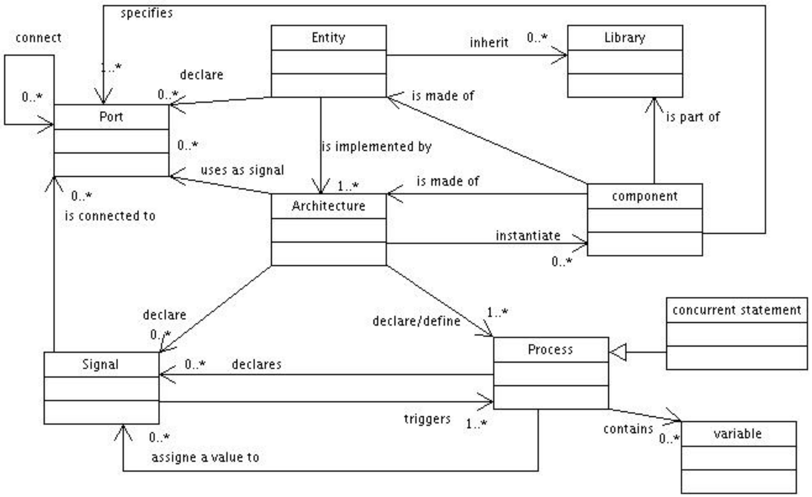
Viewpoint

Model UML-VHDL

Meta-Modelling



SystemC (2.0)



VHDL

Mappings

- **Syntactic Mappings**
  - cf: UML Class =
    - C++ Class, Java Class, Eiffel Class, C-struct VHDL process
    - etc...
- **Semantic Mappings**
  - UML Class + State + Interaction =
    - configurable template structures
  - Semantic equivalences between different languages
- **Under MDA, Semantic Mappings will become more prevelant than traditional syntactic mappings**

Issues

- **Top down development**
  - modelling and configuration management become critical

- **Mappings become highly complex**

  - Emphasis shifts from modelling to producing mappings
  - Libraries of mappings need to be created
  - Growth industry:
    - look out for mapping conferences, mapping patterns etc

- **Bottom-up modelling...**
  - especially in embedded systems:
    - hardware/software constraints
    - specific platforms
    - many semantics, esp: concurrency

DSP Development with MDA

DSP in Mobile Devices

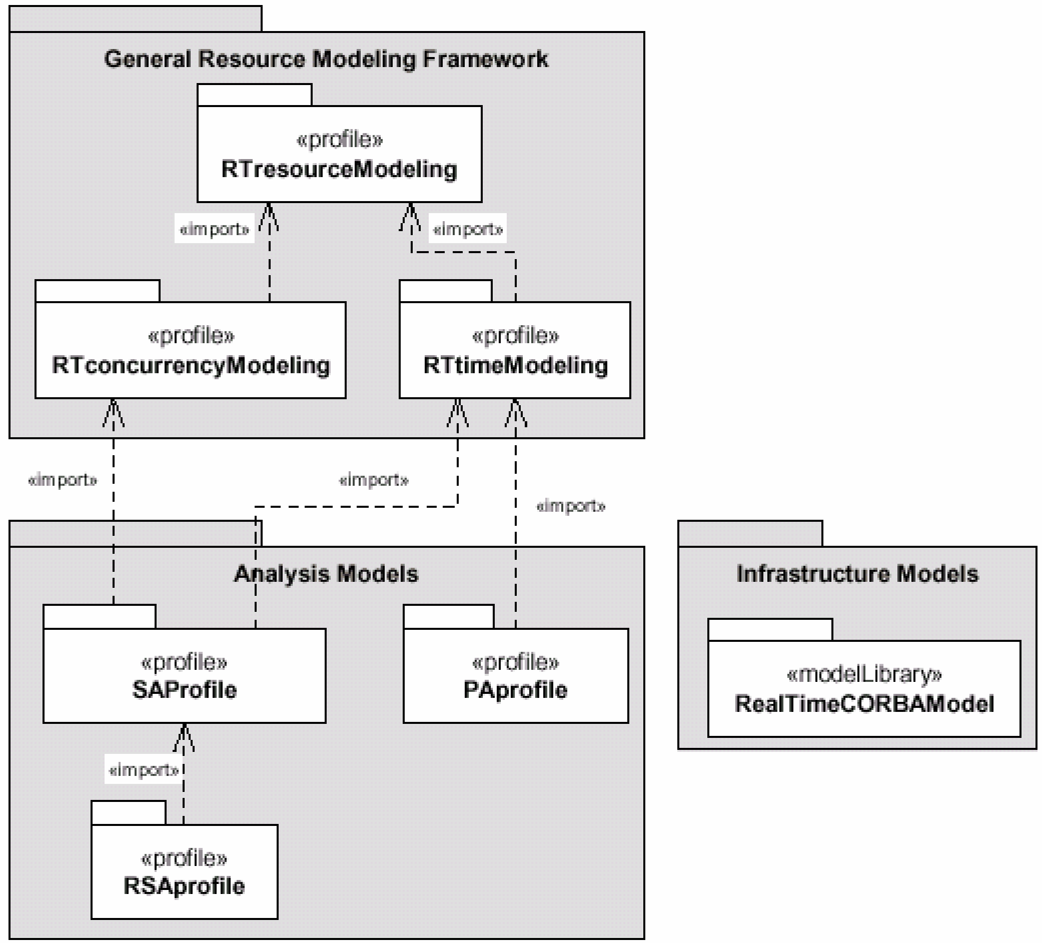
- mixture of Soft & Hard-real time
- Many concurrent processes
- Many functions
  - GSM
  - GPRS
  - CODECs
    - Audio
    - Mpeg
    - MIDI
  - Image procesing
- Close to hardware
  - software vs ASIC
- Power consumption

- Devices are (at present!) dual processor:
  - MCU
  - DSP
- Move towards custom ASIC in some devices
- Multiple DSPs

UML-RT

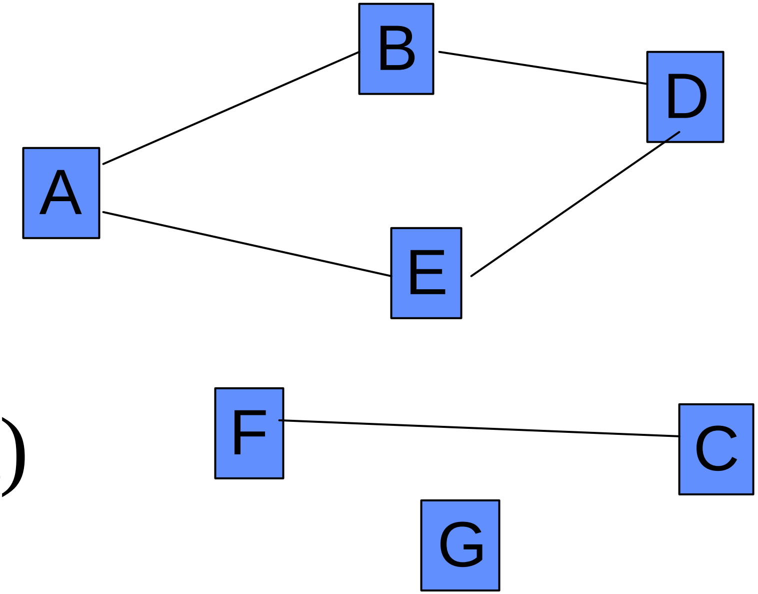
- A Profile based upon UML 1.3/UML 1.4

- Standard defintion of time
- Quality of Service
- Concurrency defintions
- Active/Passive classes
- Signals & Port



Rate Monotonic Analysis

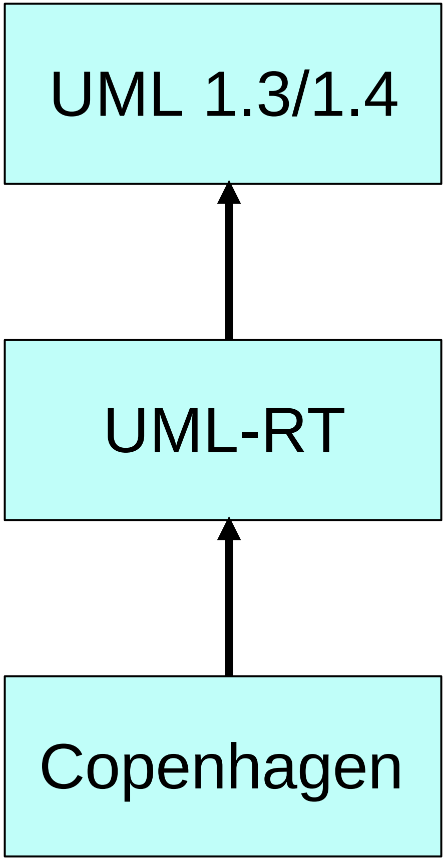
- Formalism for estimating schedulability of a set of tasks with dependency graphs
- Worst Case Analysis
- Extended to take into consideration:
  - Different scheduling executives
  - Priorities
  - Priority Inversion



$$\sum_{i=1..n}\left(\frac{Ci}{Ti}\right) < n(2^{1/n}-1)$$

Copenhagen

- Adds to the following
  - Class diagrams
  - Sequence
  - State
  - Deployment
- Includes
  - Threads
  - Modes
- Tool Support
  - Collaboration Diagrams

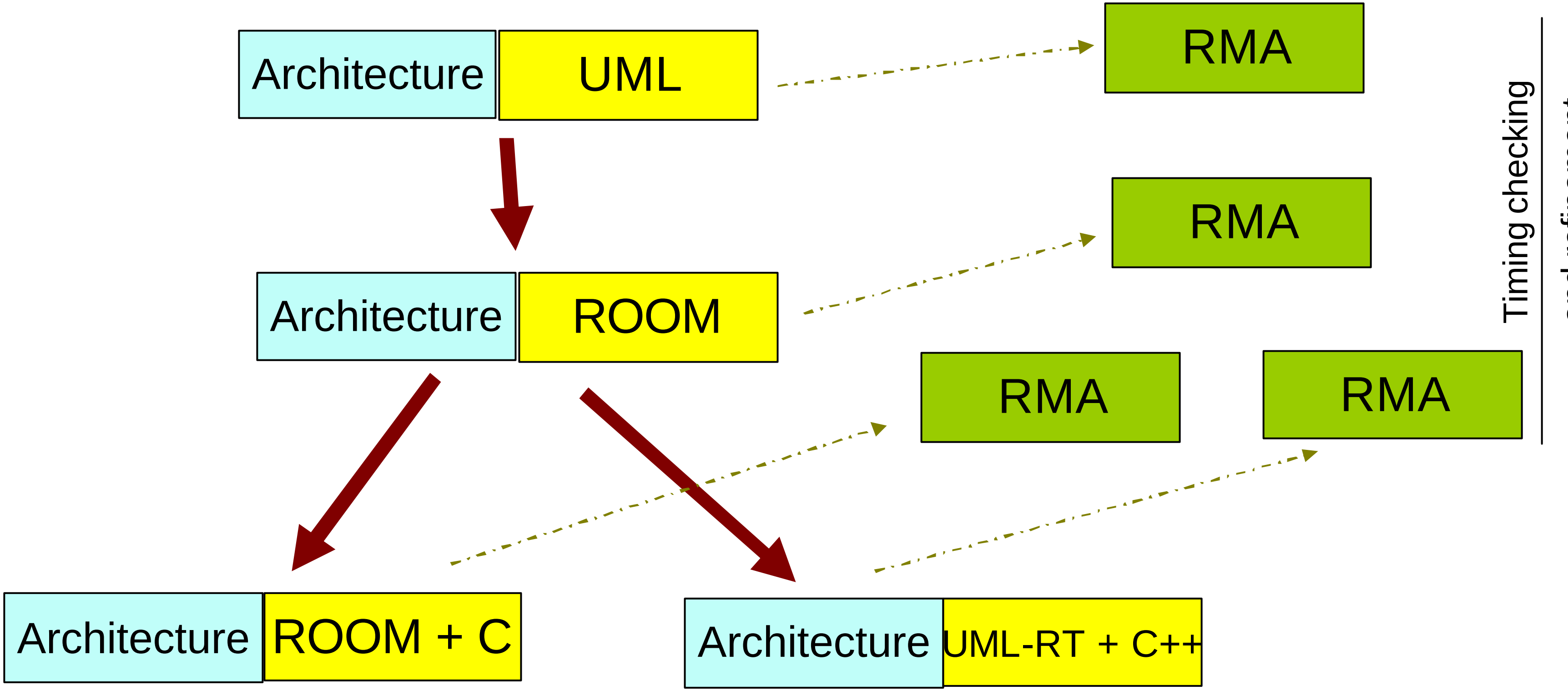


UML 1.3/1.4

UML-RT

Copenhagen

MDA in DSP Development



MDA first impressions

- MDA at this time (DSP development) was different but mappings, meta-models and experience lacking

- Tool support was non-existent
  - mappings by hand
  - XML/XMI doesn’t work between tools

- We did get:
  - better ideas on how to perform MDA development
  - a much better set of models
  - much more tracability with regards to
    - structural properties
    - behavioural properties
    - timing properties

PUSSEE

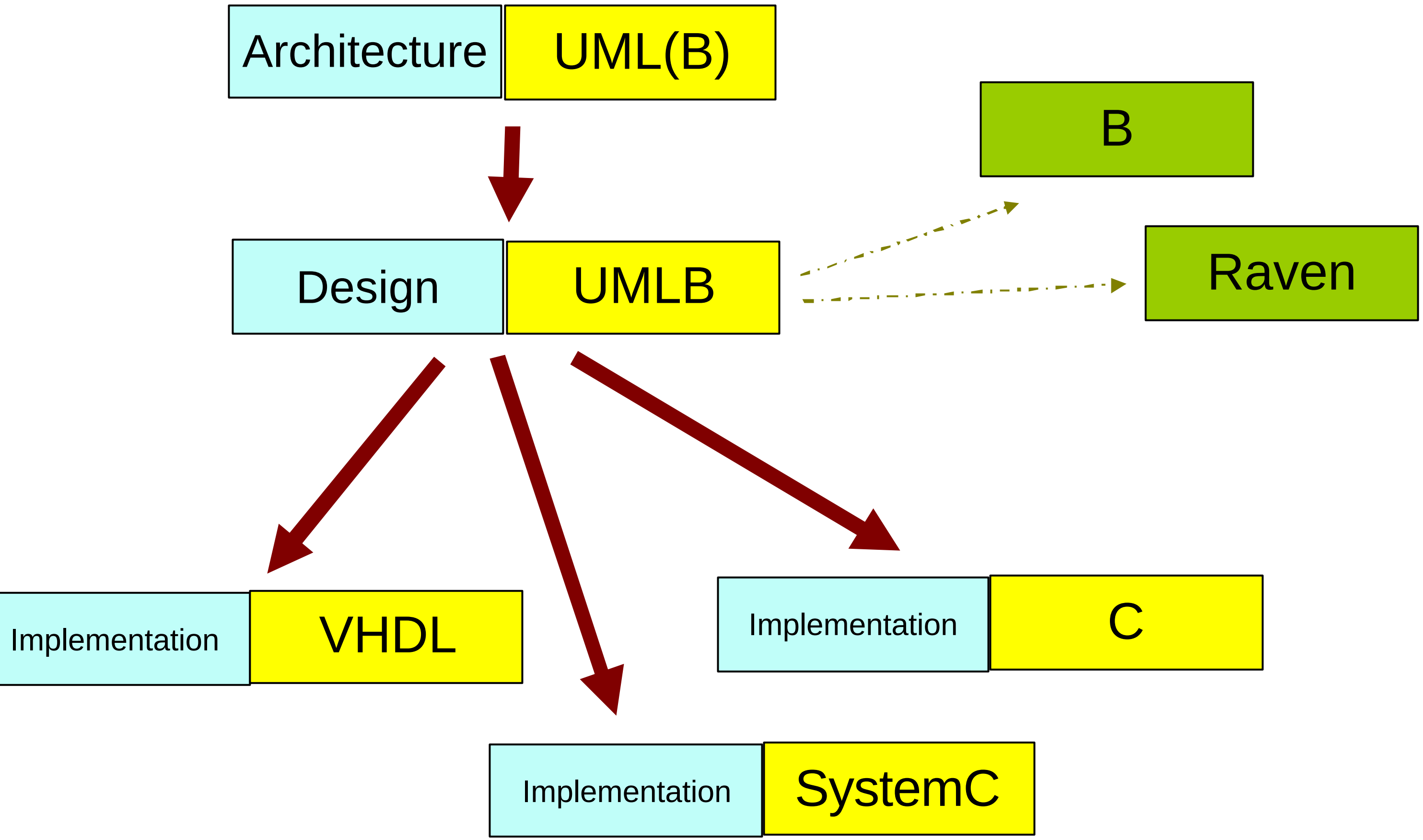
PUSSEE

- Effort to integrate UML and B
- Formal design of software and hardware components
  - generates libraries of proven components

- B to VHDL/System C mappings
- UML+B to RIL/Raven (model checker)

- Fits well with the “MDA ethos”

PUSSEE MDA Development



Architecture UML(B)

B

Design UMLB

Raven

Implementation VHDL

Implementation C

Implementation SystemC

Development in PUSSEE

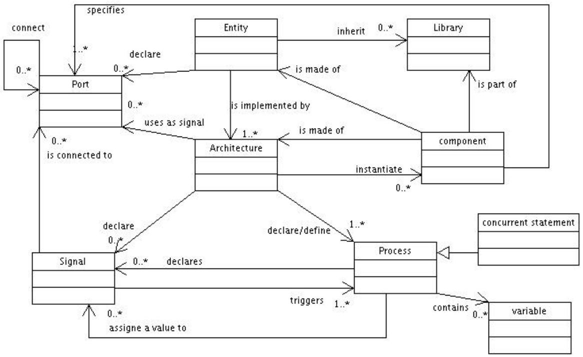
- As development proceeds the underlying language changes
  - UML
  - UML-RT
  - UML-B
  - UML-Hardware Profile
  - UML-VHDL
  - UML-SystemC
- Each level builds upon the features of the previous level
- Mappings are developed between each pair of languages
  - UML-B to VHDL
  - UML-B to SystemC

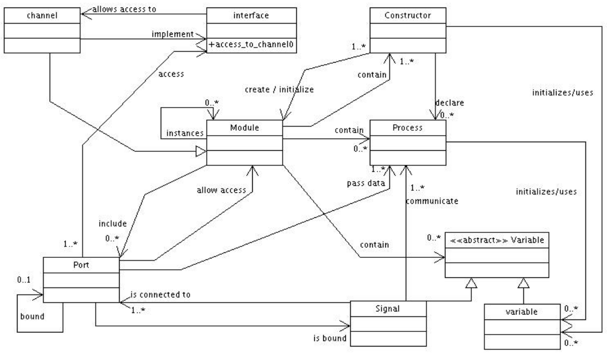
- UML is used as the carrier language supplimented with semantics and notation from
  - B (B-Method)
  - VHDL
  - SystemC
- B becomes a platform independent language to generate VHDL, SystemC C etc

Meta-Models support PUSSEE

- **Primarily VHDL and SystemC**
  - suppliments UML with concepts from both VHDL and SystemC

- **UML Hardware profile**
  - set of UML profiles targetted towards software-hardware development
  - a generic UML profile constructed from the commonalities between existing hardware description languages
    - VHDL, SystemC, Verilog etc





PUSSEE Conclusions

- **Project still underway**
- **Process is a difficult point**
  - B-Method employs strict refinment
  - UML suggests a more exploratory style
  - VHDL/SystemC ideas need to be incorporated earlier into the process
  - Purely syntactic mappings from B to VHDL/System C produce large pieces of hardware...
- **MDA Mappings are still informal**
  - Tool support for MDA is still poor
    - despite what the tool vendors say...

Conclusions

MDA’s Future

- **MDA will become the framework in which software is developed**

- **MDA requires much more formal work**

  - mappings are still informal and relatively weak (syntactic in nature)
  - use of evolving underlying languages is difficult
  - use of platform independent models and differentiating between what is platform independent and what is platform specific is difficult
  - processes and methodologies are not ready for MDA

- **Tool Support**

  - does not exist

Integration

- **DSP**
  - UML-RT to C mappings
  - UML-RT to RMA mappings
- **PUSSEE**
  - UML+B
  - UML+B to B
  - UML+B to VHDL/SystemC
  - UML+B to Raven
- **Combination...**
  - UML+B to RMA mapping
  - UML+B to Raven provides basis for
    - UML to SMV, UML to <<*some model checker*>>

and finally...

- MDA is to be driven by “real” concerns
- Progress only made through “real” use
  - libraries of mappings
  - meta-models => languages
  - experience
- “Everything” is MDA