EE249 Discussion Session

Safa Messaoud
Antonio Iannopollo

Quo Vadis SLD? Reasoning About the Trends and Challenges of System Level Design

By Alberto Sangiovanni-Vincentelli

Outline

- Motivation for System-Level Design
- Platform Based Design
- STATE-OF-THE-ART in embedded system design review using the PBD paradigm
- Metropolis Framework
- Conclusions

Motivation for system Level design

1. Rapidly increasing complexity

Function diversity

Supercomputer compute performance

Industrial Control
High reliability
Real-time capable
Embedded SW
Safety criteria
Efficiency

Can traditional design flows (i.e. RTL) continue to meet these demands?

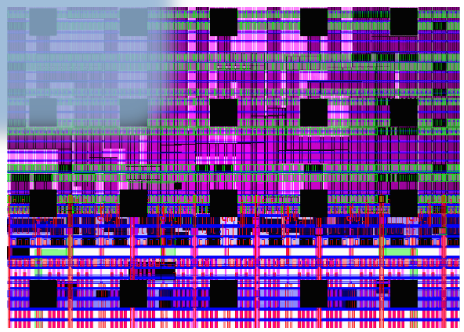
Integration Complexity
"Plug & Play"

Costly!

```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author john doe <doe.j@example.com>
 */
HelloButton()
{
  JButton hello = new JButton( "Hello, wor
  hello.addActionListener( new HelloBtnList

  // use the JFrame type until support for t
  // new component is finished
  JFrame frame = new JFrame( "Hello Button"
  Container pane = frame.getContentPane();
  pane.add( hello );
  frame.pack();
  frame.show();           // display the fra
}
```

Decreased productivity
of SW Programmer
(redesign + long verification time)



High Integration
(Moore Law)

System Level Design

The process to implement a desired function with a given set of physical components

- Holistic system considerations (HW&SW) and decision making on important design criteria early in design process
- Possible validation of important aspects of system behavior during early phases of design process
- Design reuse → Reduction of validation effort & costs
- Better overall understanding of the interplay of subsystems originating from different suppliers

+Motivation for system Level design

2. Industrial supply chain landscape

Mobile Communications Design Chain

Application Developers

at&t alltel wireless T··Mobile··· verizon Sprint

SIM Card locks the device to one Service provider

Service provider

Device Maker

SAMSUNG

IC provider provide IP to multiple device makers

Outsourcing Companies

IP Provider

+Motivation for system Level design

2. Industrial supply chain landscape

Automotive Design Chain

- Standards for interoperability among IP and tools

⟶ reducing Costs &Time to Market

- Hard real-time software is difficult to share !

Car Manufactorers (OEM)

Tier 1 Supplier

Tier 2 Supplier

Manufactoring Supplier

Further Challenges

Boundries between companies are not clean
(Misinterpretation of the design specs)

Non functional Specs are difficult to trace

What is a Platform?

IC Domain

- "… a flexible integrated circuit where customization for a particular application is achieved by programming one or more of the components of the chip ."

PC Domain

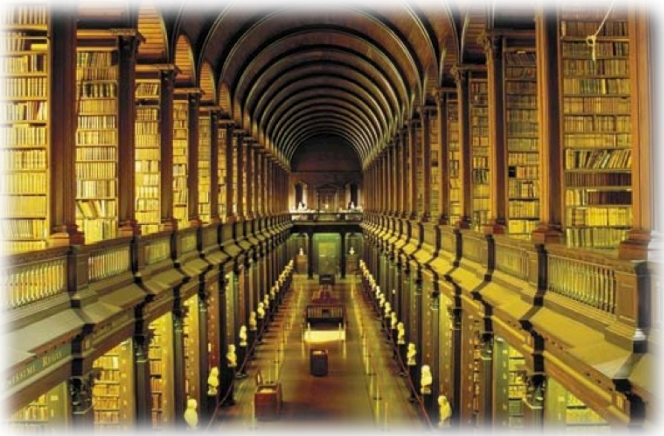
- Instruction set
- Buses
- A common set of I/O devices

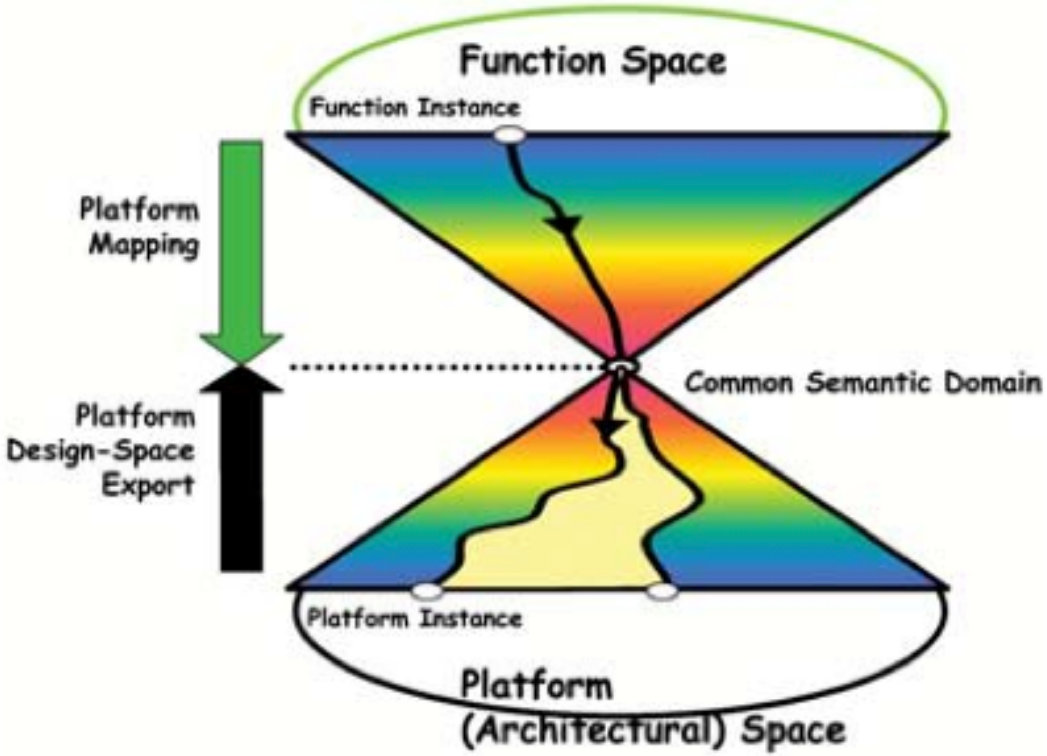
System Domain

- Loose definition: "something" that allows to develop quickly new applications

PBD:
Platform Based Design

- A platform can be viewed as a library of components
- Both Computational and Communicational blocks
- Each element can support a (set of) functionality with some performance constraints
- A platform instance is a set of components selected from the library
- It is the how the system does a particular functionality



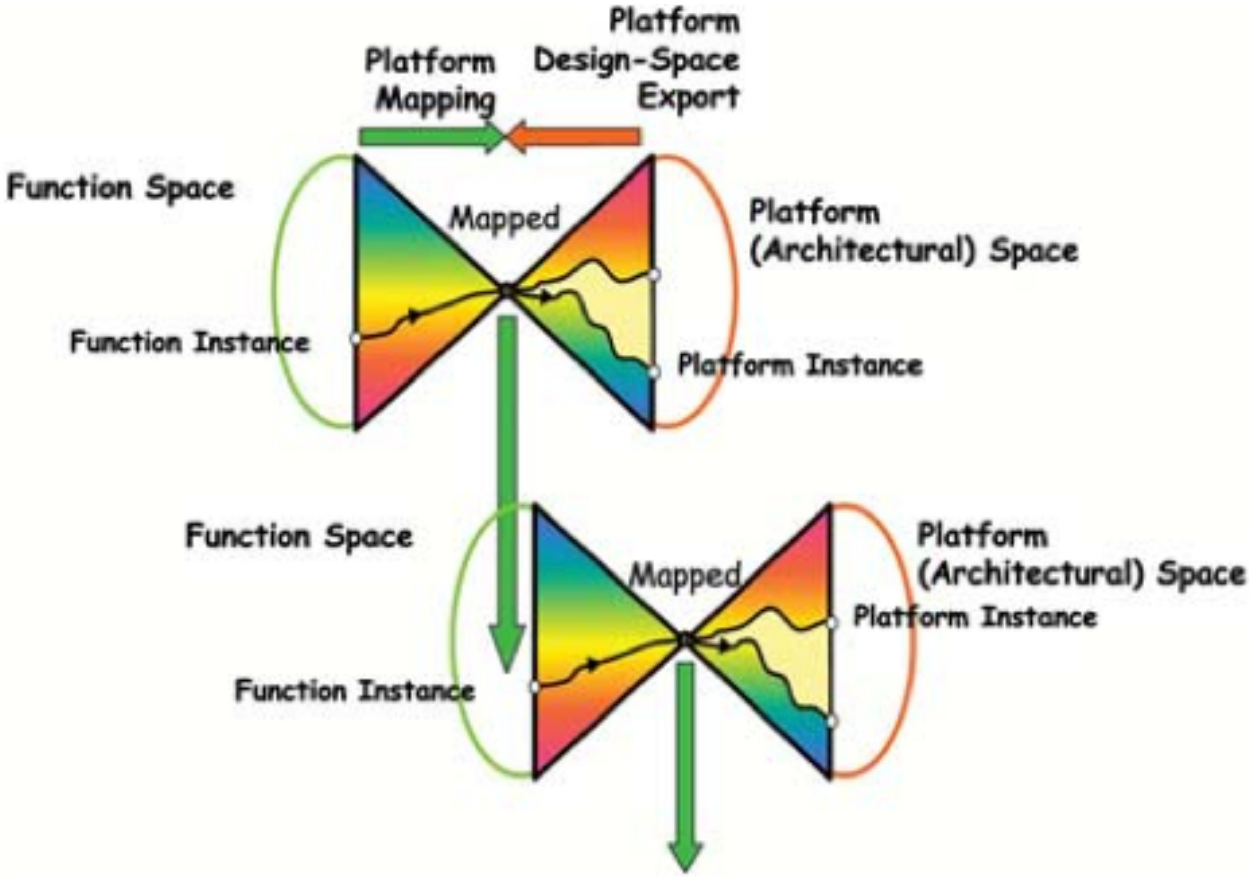


Meet in the middle

Top-down: application view

Bottom-up: architectural view

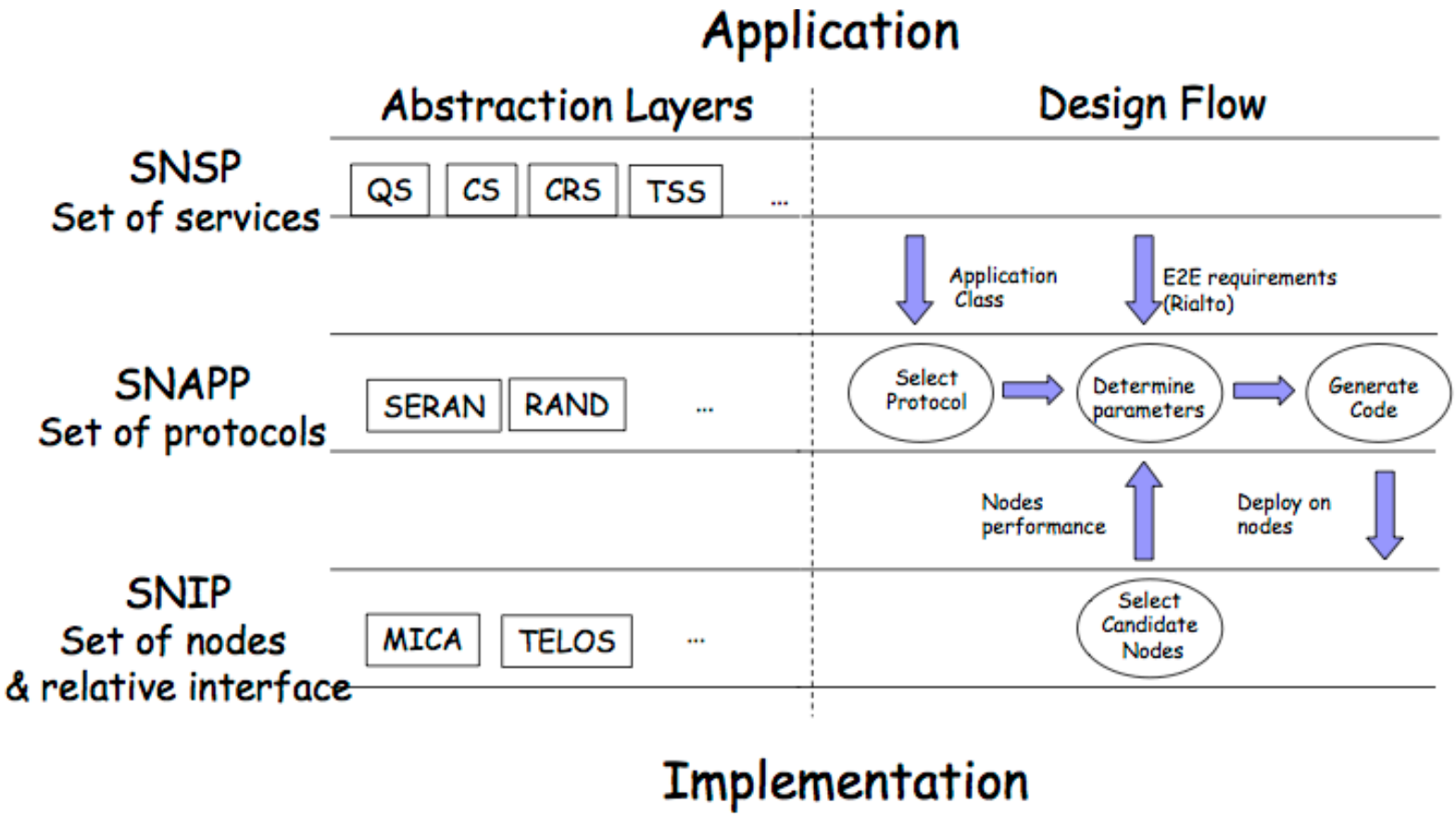
The middle represents the mapping between functionalities and a platform abstraction



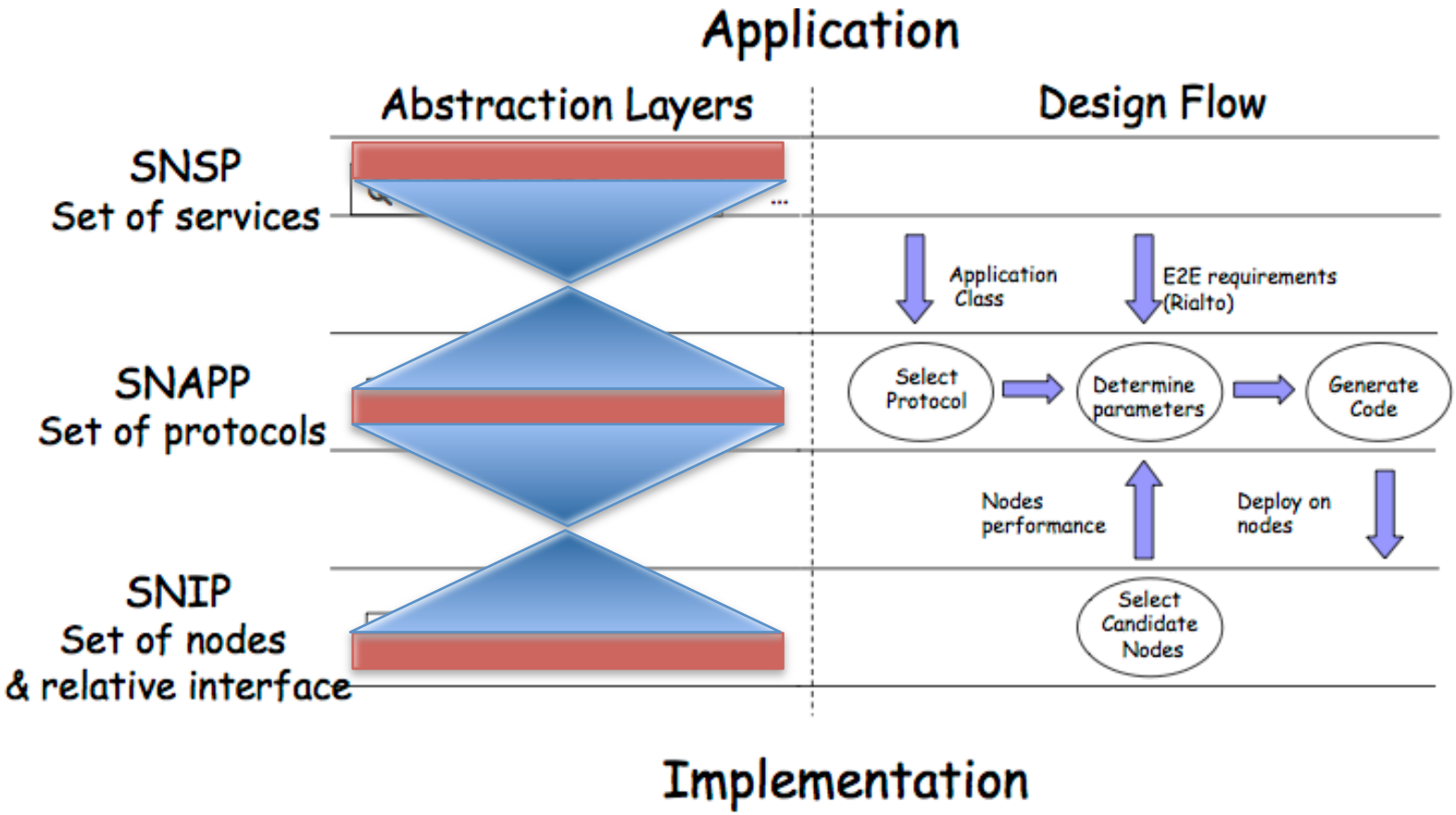
Fractal nature of design

- Result of the mapping process
- Refinement process
- Several iterations until components are described in their final form
- Use of intermediates platforms
- Avoid large loop iterations

An application: Wireless Sensor Network

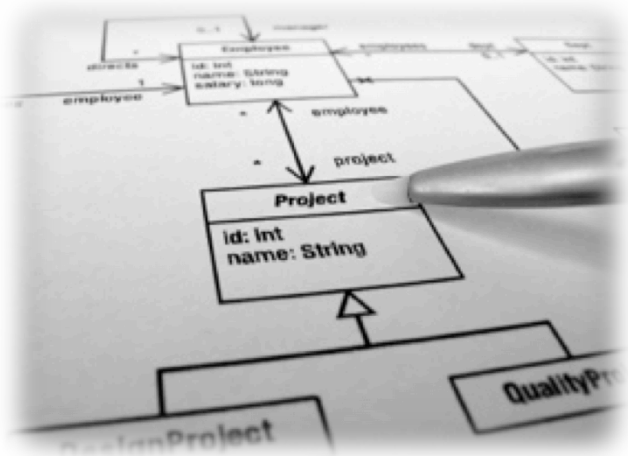


An application: Wireless Sensor Networks



PBD ~= Model Driven Development ?

- Concept of Platform 
- Separation between functionalities and architectures 
- Model transformations (some tools) 
- Detailed description of platform and "semantics" for embedded SW design 



PBD key points

- Platforms as a contract
  - Useful for managing inter-actor communication
  - Prevents long design cycles
- It helps to raise the abstraction level of the design process
- BUT
  - A specific training is required
  - It needs the presence of adequate tools

STATE-OF-THE-ART in embedded system design review using the PBD paradigm

1. Functional aspects

Description of the functionality through programming languages using Models of Computation

- Languages for HW Design
- Languages for SW Design
- Models of Computation

Languages for HW Design

- SystemC
  - A class library of C++
  - Concurrency Implementation
  - Mainly used for Simulation
- SpecC
  - Super set of ANSI C
- System Verilog
  - Build open HDL
  - Does all what SystemC can do

Languages for SW Design

- Main difficulty: Programmer productivity & design correctness
- Use of Synchronous Languages
  - Strong formal semantics
  - verification and the code generation problem easier by construction
  - Separation between computation and communication
  - Application in safety critical domains (Aviation, Automotive)
  - Esterel, Lustre, Signal

Models of Computation

- Choice of the Model class depends on the system properties
  - Discrete time model → Flexibility
  - FSM, Data Flow Graph → less flexible, easier to analyse and synthesize

- Heterogeneous Models of Computation
  Numerous approaches for the interaction model
  - LSV Model
  - Interface Automata

- Environments for Heterogenous Models of Computation
  - Ptolemy II
  - ForSyDeand SML-Sys
  - Simulink
  - LabVIEW

STATE-OF-THE-ART in embedded system design review using the PBD paradigm

2. Architectural aspects

Architecture: Netlist that establishes how a set of components is connected

SW Architecture description

HW Architecture description

SW Architecture Description

- Unified Modeling Language UML
  - Successive refinement approach to SW design
  - Fuses the concept of visual languages with the one of oriented languages
  - Structure based on diagrams
  - Profiles refine UML for specific applications
- Eclipse
  - Open source platform
  - Contains a plug-in Java-based environment for building Software

HW Architecture Description

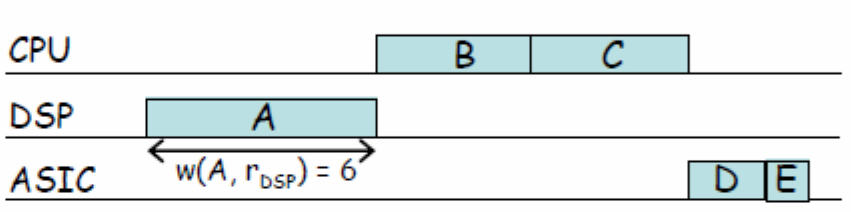
- Useful when providing a model for the execution so that the performance and the properties can be analyzed
- Transaction Level Modeling TLM
  - Levels above RTL
  - TLM 2.0: the most recent version
- Assembly tools
  - To explore model creation, integration, simulation and analysis
  - CoWare, Synopsys, Mentor, ARM
- Communication based design
  - Design of interconnect infrastructure and IP interfaces
  - Trend: Global Interconnect

HW Architecture Description

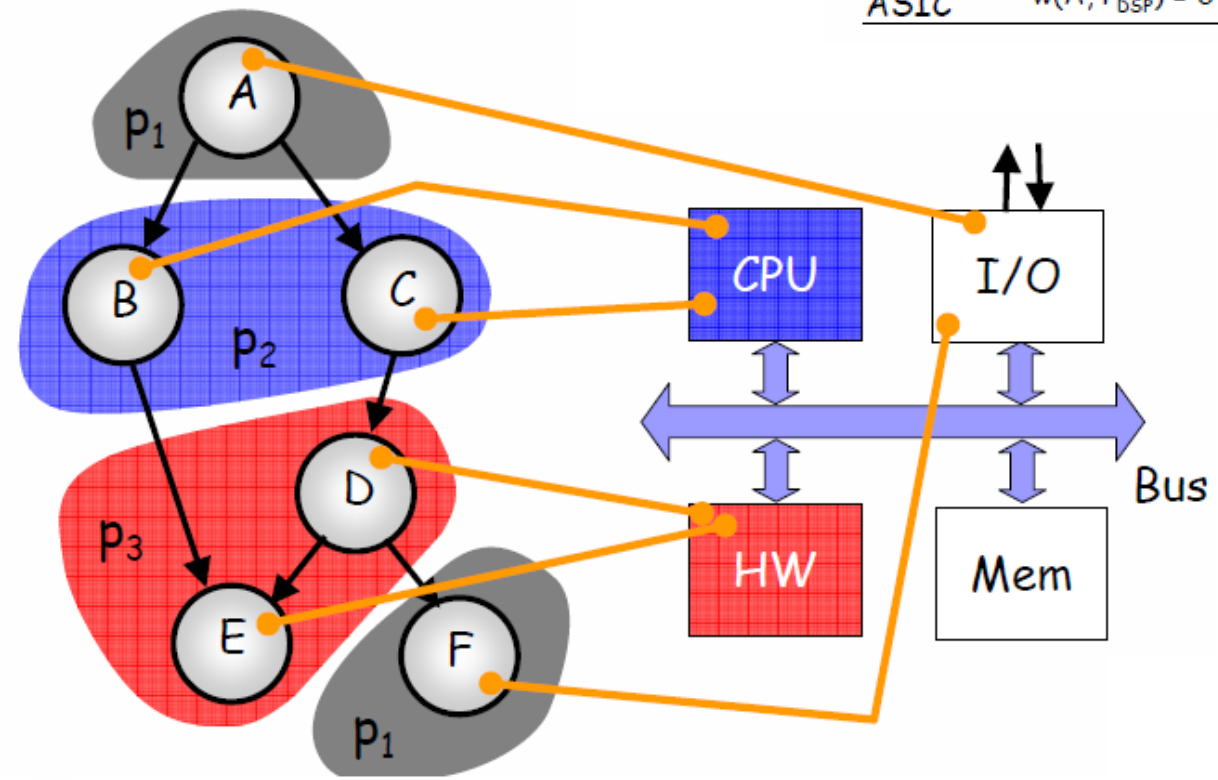
- Microprocessor Modeling
- Embedded systems contain software programmable processors
  - Virtual Processor Model
  - C-Source Back Annotation
  - Interpreted Instruction-Set Simulator
  - Compiled Code Instruction-Set Simulator
  - Worst Case Execution Time Estimation

STATE-OF-THE-ART in embedded system design review using the PBD paradigm

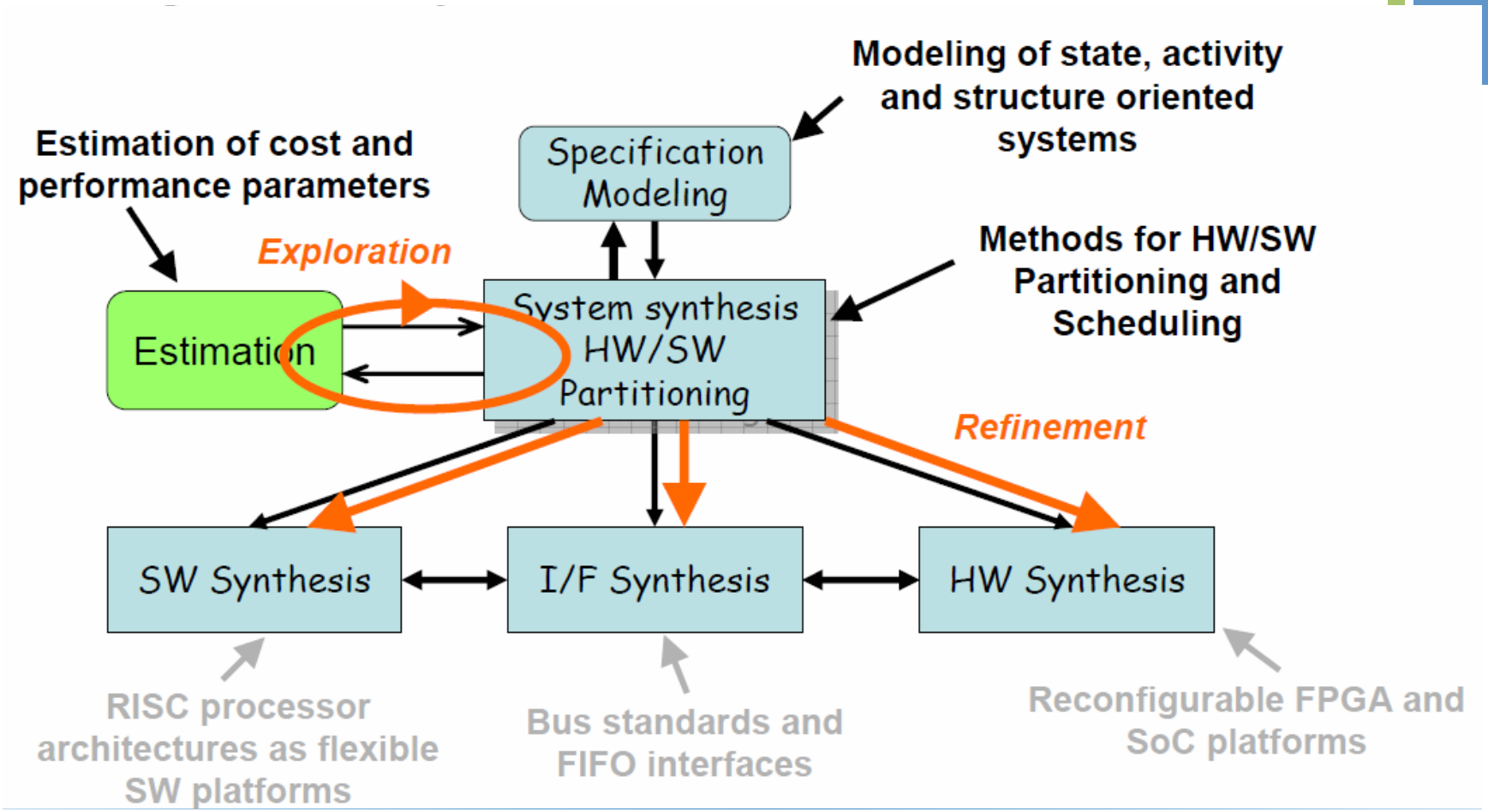
3. Mapping



Sheduling



Design Flow at System Level



CoFluent Studio

cadence™

MATLAB®
SIMULINK®

MESCAL

CoFluent Studio

Metropolis

SIMULINK

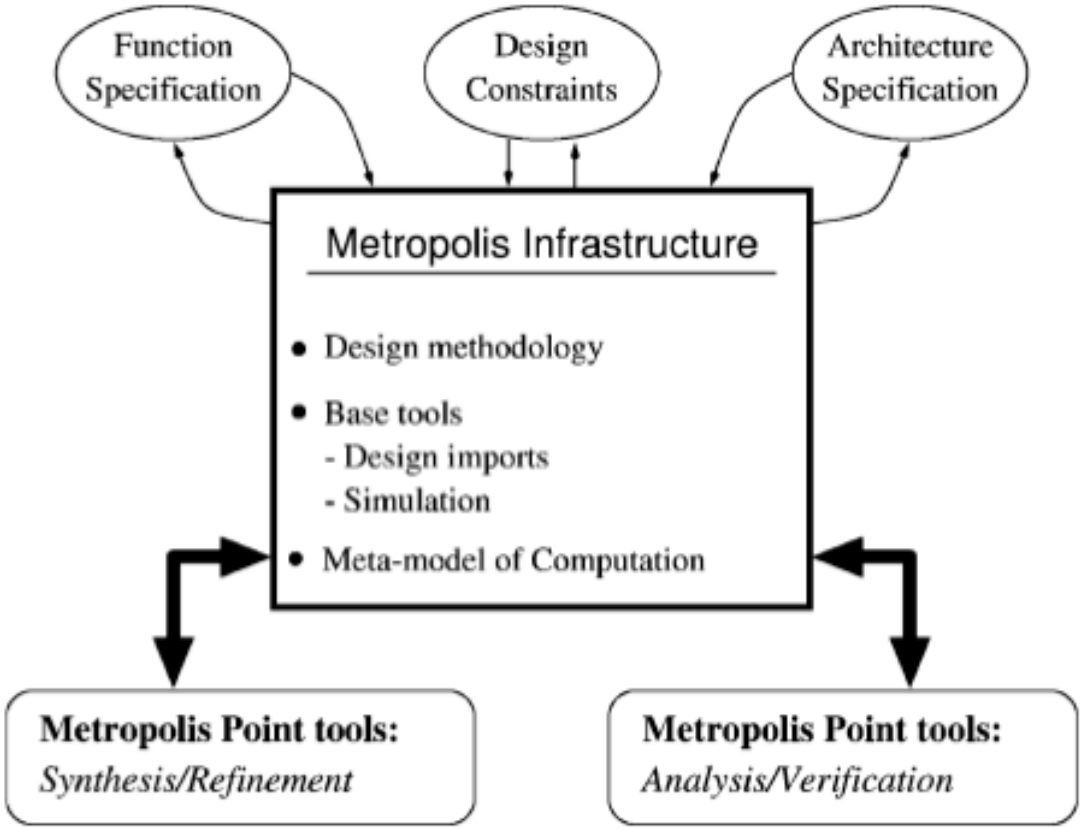
MESCAL

Metropolis

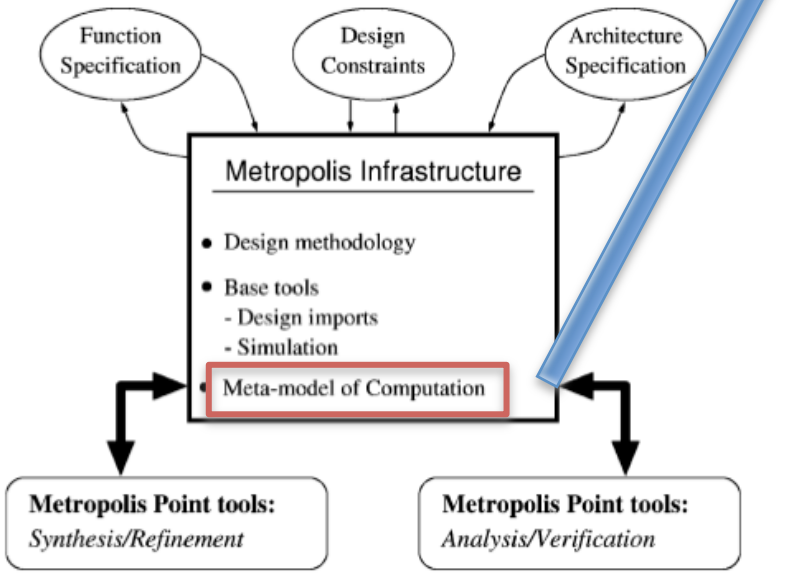
Framework

- Can manage different models of computation and different layers of abstraction
- Can analyze pure functional designs, as well as mapped designs
- Providing algorithms and tools for all possible design is not the primary goal
- It offers a consistent vision of the design process
- Domain specific algorithms can be "plugged-in"

Metropolis Architecture



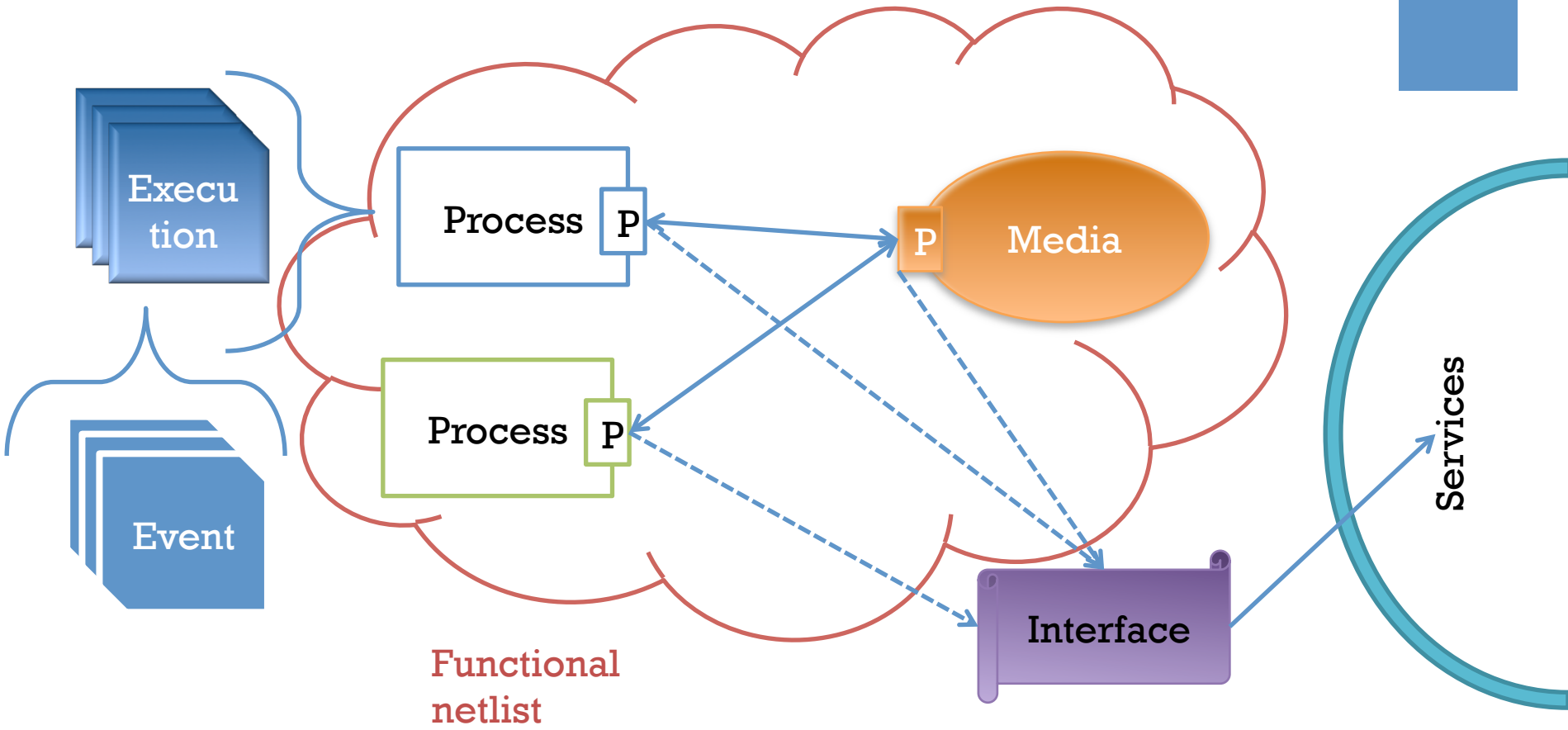
Metropolis Architecture



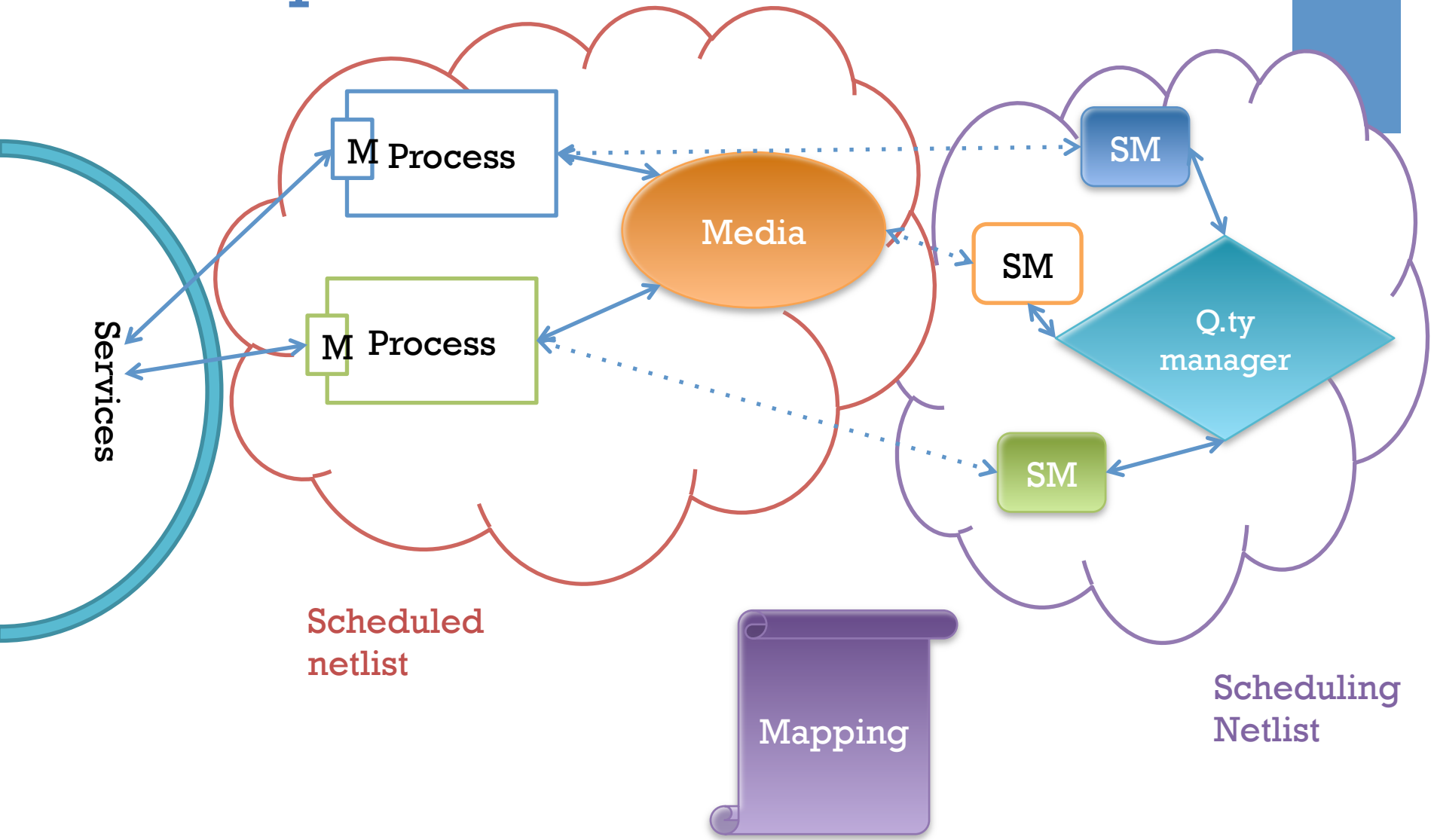
Metropolis Meta-Model (MMM)

- Powerful enough to represent common used MoC and concrete formal languages
- It "works" for functionalities, architectures and mappings
- Functional requirements described by processes, communication media and netlist as I/O relations (or algorithms)
- Constraint over quantities described by temporal and propositional logic
- Processes and media can be replaced by subnetlists

Metropolis: functional model



Metropolis: architectural model



Conclusions

- System Level Design is advantageous in terms of efficiency and productivity
- Platform based design is a powerful concept
- For some aspects, as the platform layering, a certain expertise is required
- It could be hard to apply in reality without a specific training
- System Design needs more support in terms of automated design tools
- Metropolis Framework tries to support the application of PBD
- Metropolis seems to need a more powerful human-machine interface