Metropolis: Design Environment for Heterogeneous Systems

[Abhijit Davare, Alberto Sangiovanni-Vincentelli, and the Metropolis Team]

Metropolis
www.gigascale.org/metropolis
GSRC Chess

Main Characteristics:

Platform-Based Design

Facilitate design of complex systems by orthogonalizing key aspects

Separate:
1. Functionality vs. Architecture
2. Computation vs. Communication
3. Behavior vs. Performance

❑ Platforms eliminate iterations in the design process and allow cheaper design

❑ Design space is restricted by utilizing regularity and structure

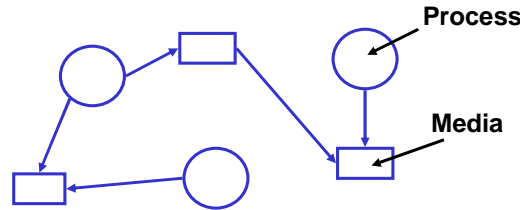❑ The number and location of platforms is often domain dependent
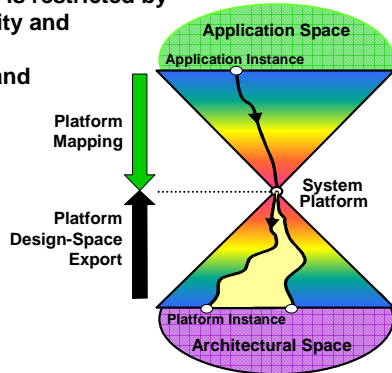
A. Sangiovanni-Vincentelli, *Defining Platform-Based Design*, EE Design, 2002

Metamodeling

The capability to describe designs with different models of computation within the same framework

- The Metropolis Metamodel specification language is based on the process networks abstract semantics

- Processes are independent threads of control within a system
- They do not interact directly, instead, multiple processes can interact with media
- Media may also interact with each other, but only under the influence of processes
- This abstract semantics can be used to model:
  - Dataflow
  - Continuous Time
  - Finite State Machines
  - And many others …

Imperative & Declarative Specification

Tailor the specification style to design needs

- Untimed sequential programs (traditional C/C++/Java) are not well suited for capturing interaction between multiple processes and the environment
- Metropolis supports the traditional imperative paradigm, but also allows declarative constraints to formally capture these complex interactions
- Declarative statements are written over events in a specified logic. An event is just a particular action carried out by a particular process
- Metropolis currently supports two different logics: Linear Temporal Logic (LTL), and Logic of Constraints (LoC)
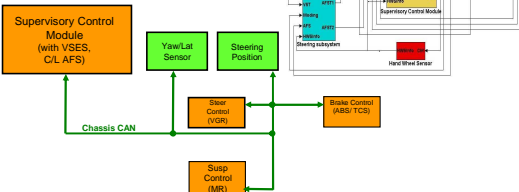- Examples:

```
E(output[i+1]) – E(output[i]) < 10
```
"Energy consumption between subsequent outputs is less than 10 units"

```
t(output[i+2]) – t(output[i]) = 10
```
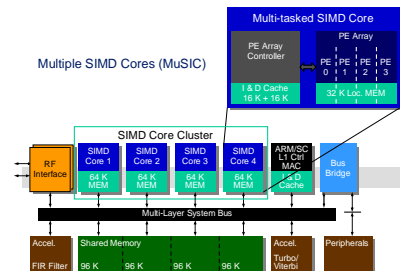"Two outputs are produced every 10 time units"

$$e_1 \Rightarrow e_2, \quad e_2 \Rightarrow e_1$$
"Event e_2 is occurs simultaneously with Event e_1"
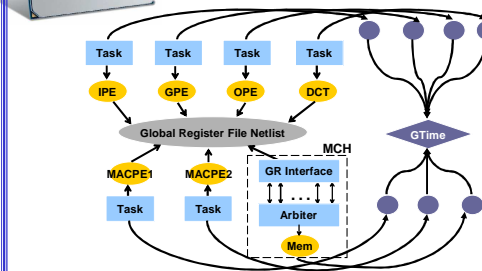
Case Studies:

GM — **Architectural Design Space Exploration and Fault Tolerant Allocation and scheduling**

Infineon technologies — **Implementation of multiple wireless protocols on an architectural platform**

intel MXP5800 DIGITAL MEDIA PROCESSOR — **Design Space exploration for imaging systems**

XILINX — **Automated characterization of architectural models for instances of the Virtex II platform**