

OMG's Third
Software-Based Communications Workshop:
Realizing the Vision

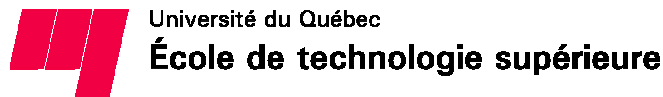
**Hardware in the Loop
Functional Verification
Methodology**

by **Pascal Giard**

Jean-François Boland, Jean Belzile

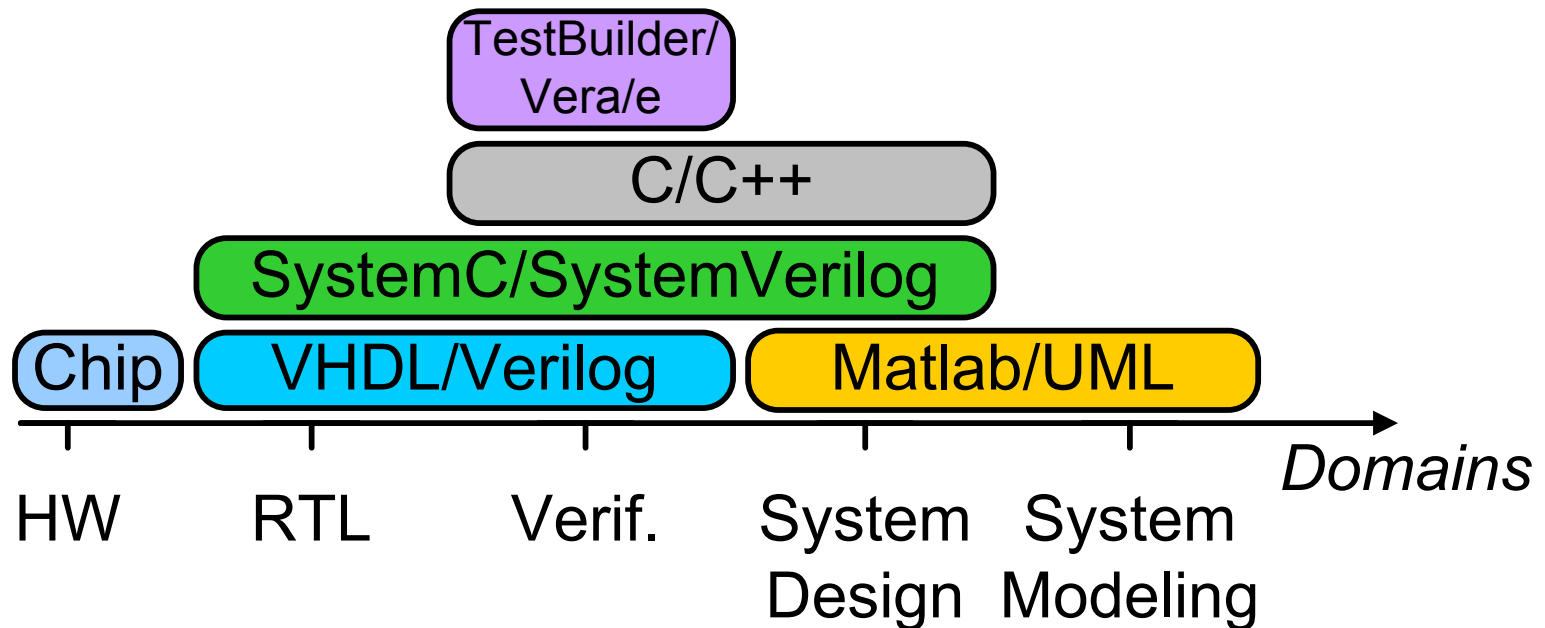
M.Eng. Student

École de technologie supérieure



Motivation: Overview

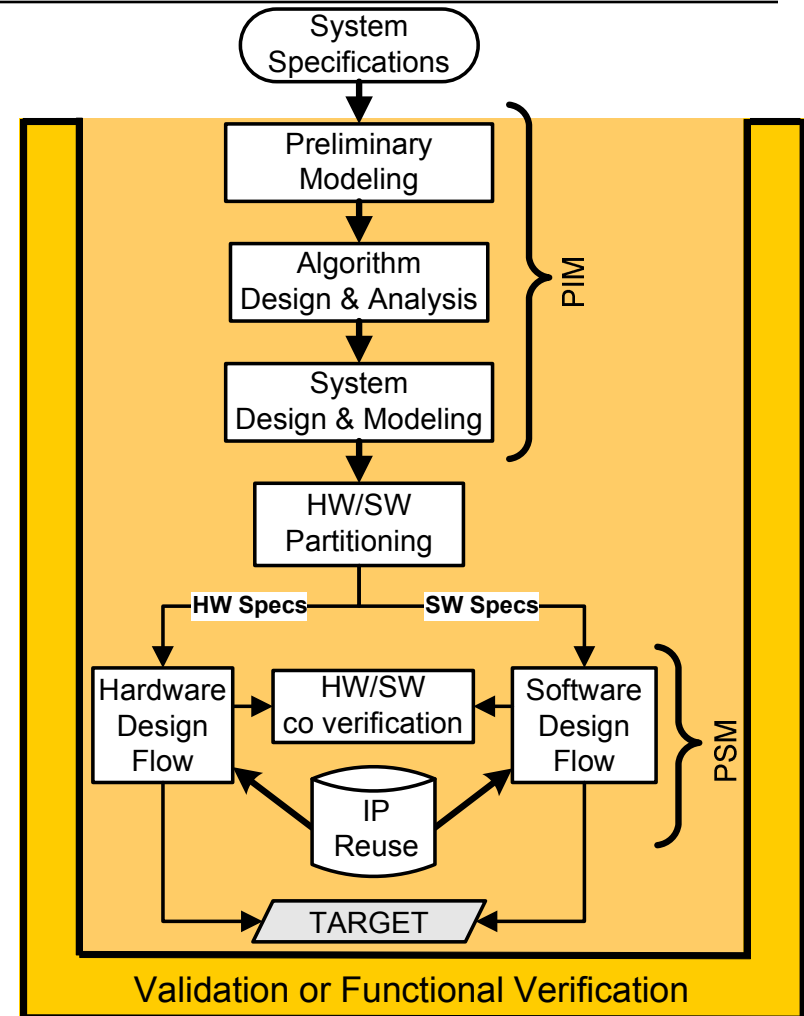
- Heterogeneous designs
 - Multiple languages, tools and abstraction levels
- Incremental design
 - Multiple refinements toward the target



⇒ **Requires co-simulation**

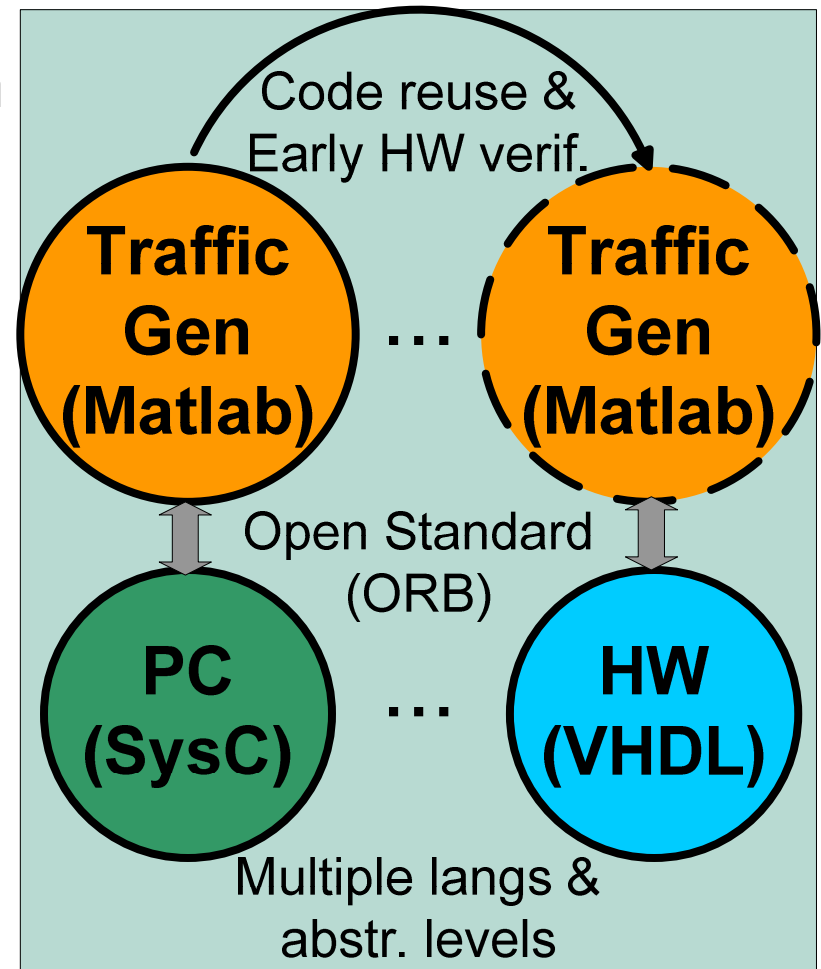
Motivation: Software Defined Radio

- Complex heterogeneous design
- Segmented implementation process
- Software Communication Architecture (SCA)



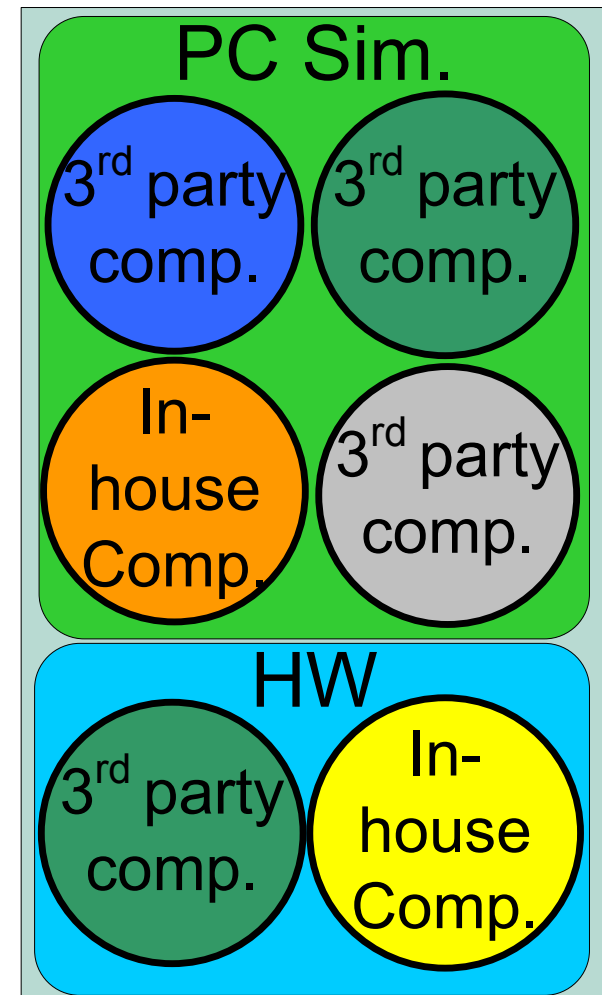
Goals: Part 1

- Reduce time spent on verification
 - Code reuse
 - Early hardware verification
 - Open standards
- Support multiple
 - Modeling languages
 - Levels of abstraction



Goals: Part 2

- Support co-simulation
 - Different languages
 - Different abstraction levels
 - Different “physical” locations
- Flexible and expendable framework:
 - Allow extensions for other 3rd party tools
 - Allow external contributions



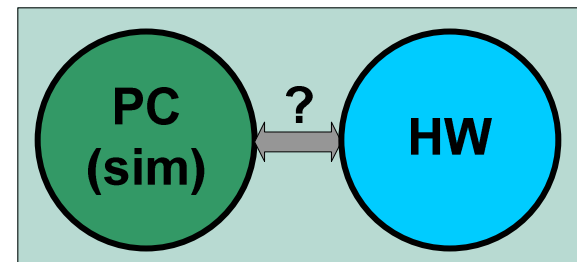
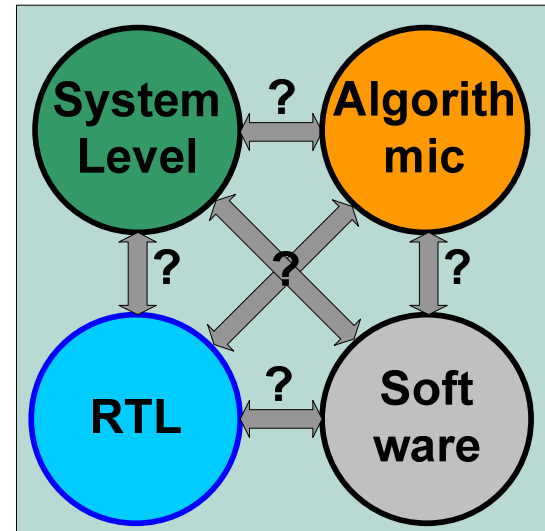


Outline

- ***Problems***
- Proposed Methodology
- Proposed Framework
- Open issues
- Conclusions and Future Work

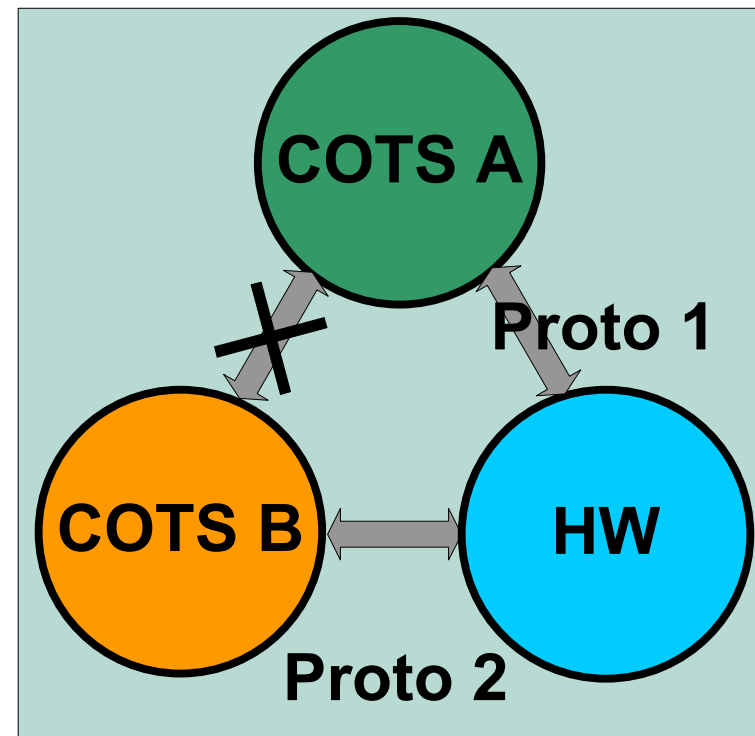
Problems: Part 1

- Core of verification: **Communication**
- Verification methodologies:
 - PC simulation
 - In-Circuit Emulation (ICE)



Problems: Part 2

- COTS tools:
 - Expensive
 - Hard to customize
 - Close standards
 - Incompatible
 - Inflexible



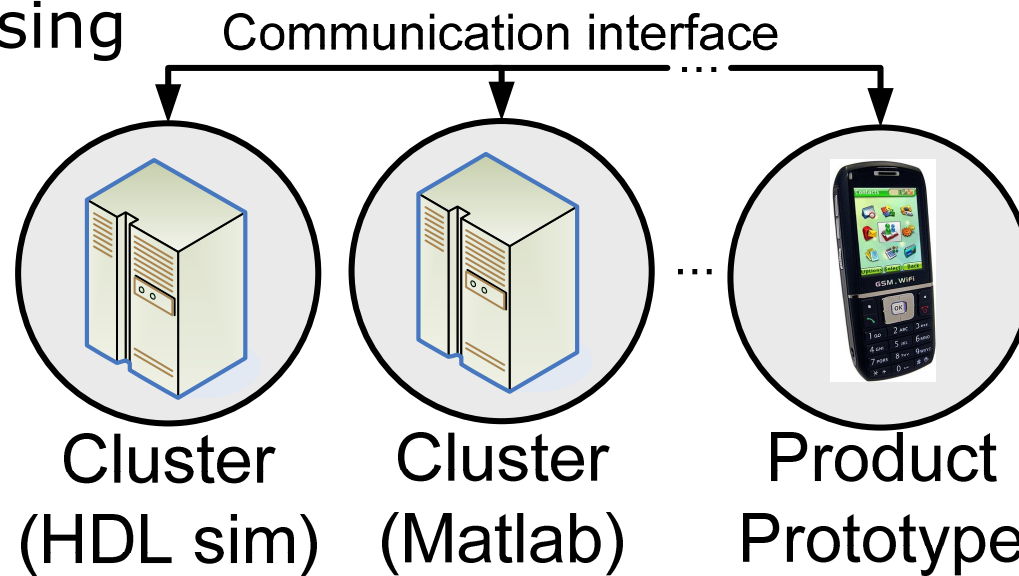


Outline

- Problems
- ***Proposed Methodology***
- Proposed Framework
- Open issues
- Conclusions and Future Work

Proposed Methodology: Part 1

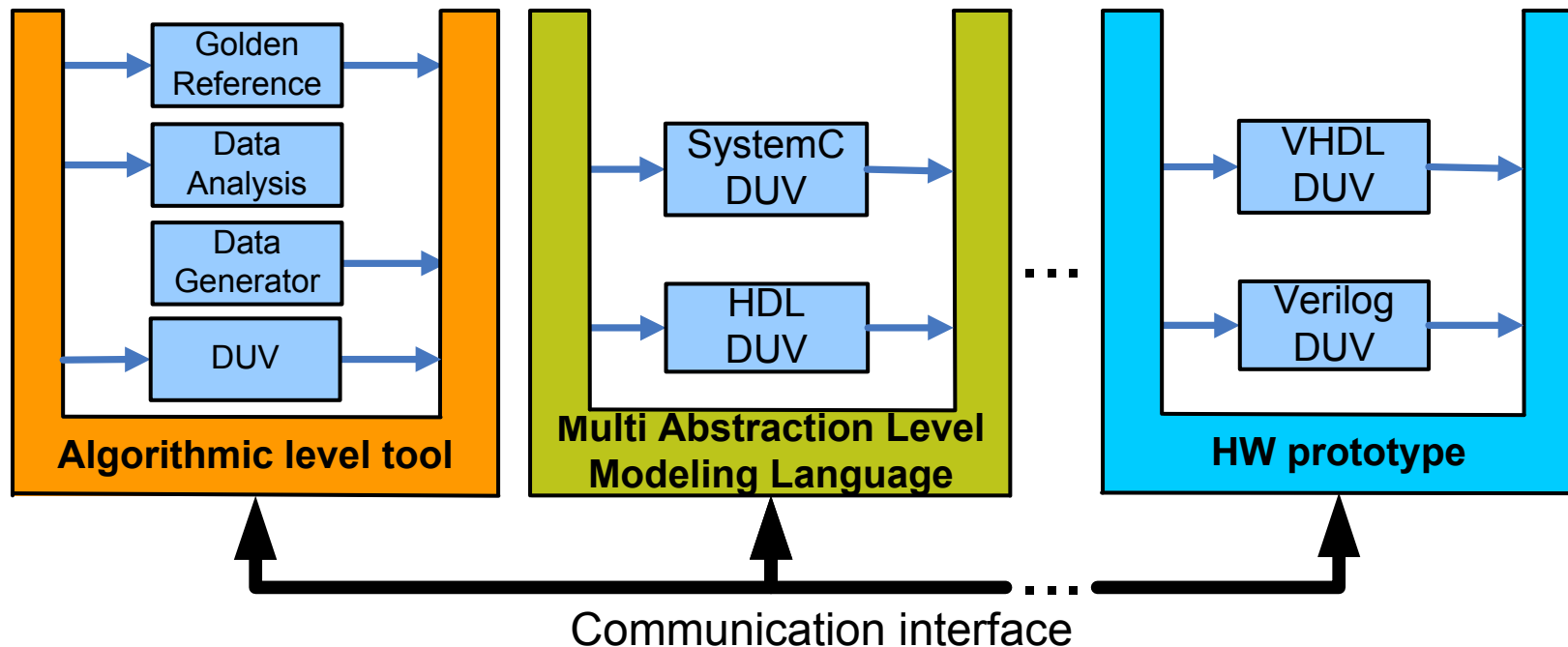
- Reduce simulation time with distributed processing



- Use traditional verification flow
 - Top-down approach
 - From specifications to final implementation
 - Multiple refinements toward target

Proposed Methodology: Part 2

- Use distributed object architecture for:
 - Verification/simulation communication
 - Internal DUV communication



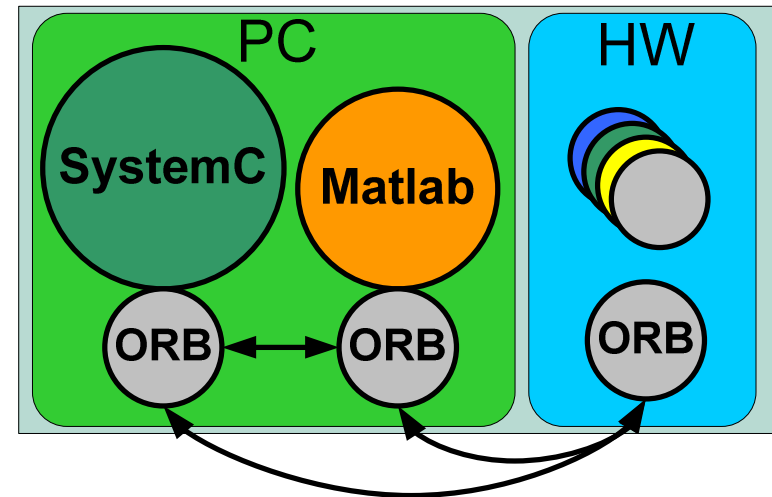


Outline

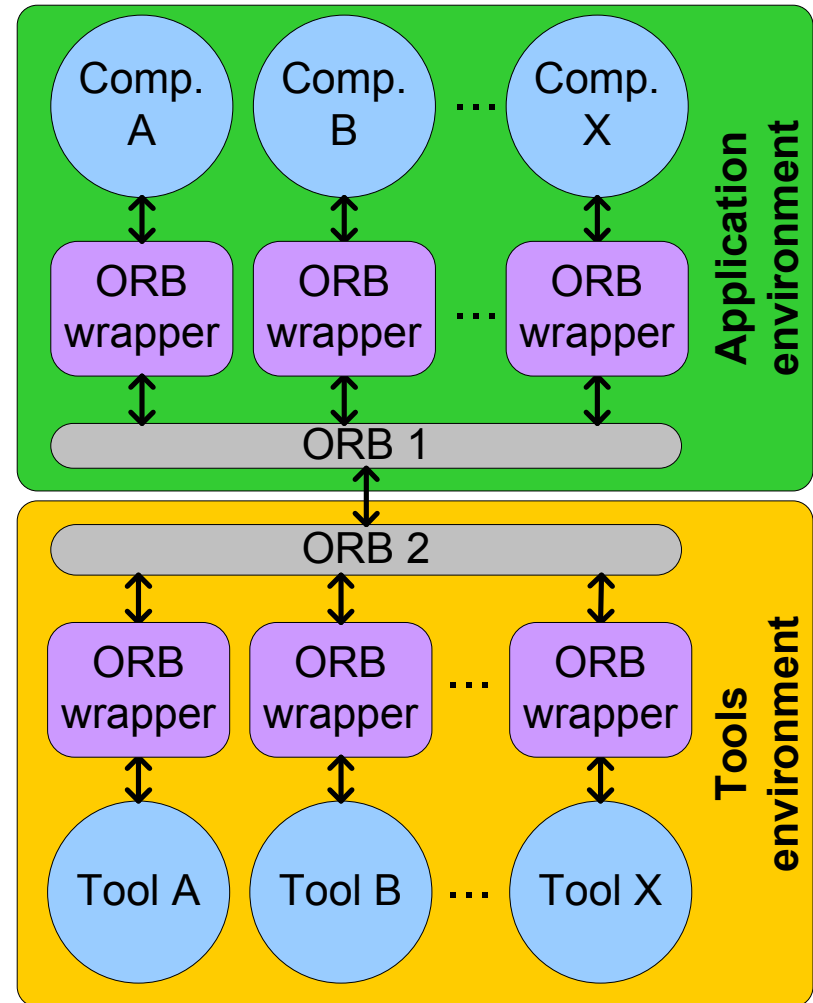
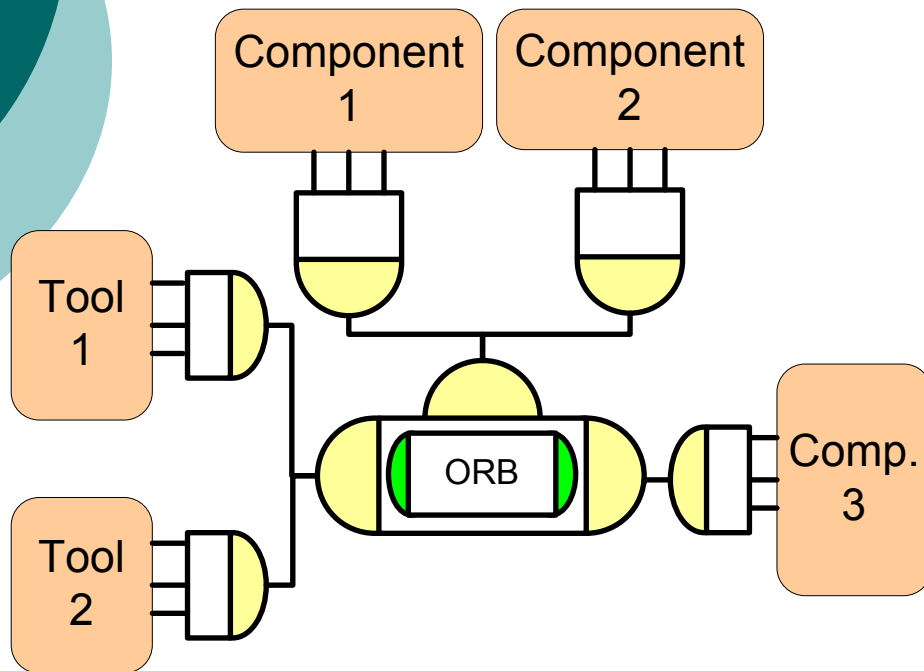
- Problems
- Proposed Methodology
- ***Proposed Framework***
- Open issues
- Conclusions and Future Work

Proposed Framework: Overview

- An ORB for everyone
- FPGA for Hardware In the Loop (HIL) verification
- 2 phases:
 - Initial version
 - First expansion

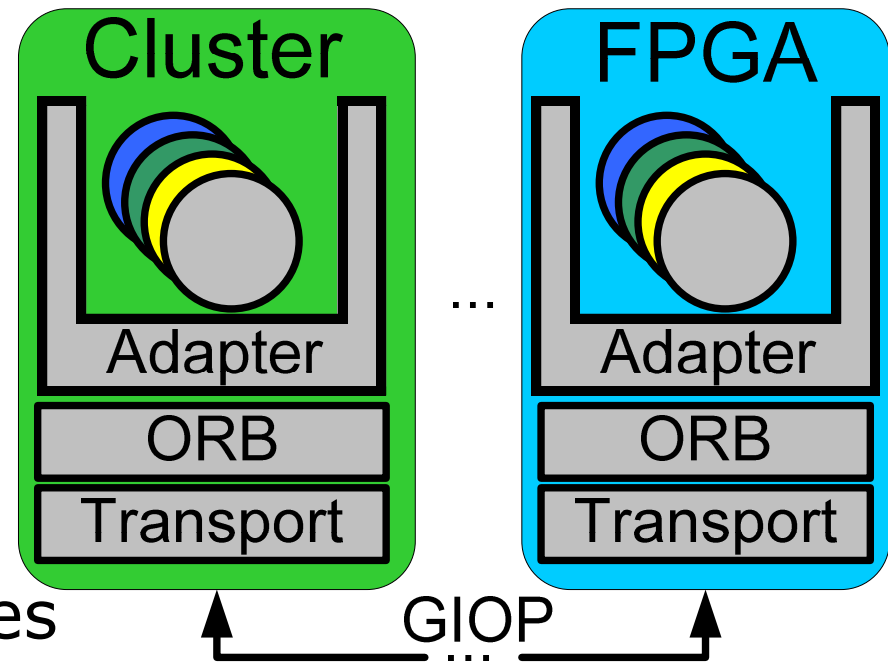


Proposed Framework: Models



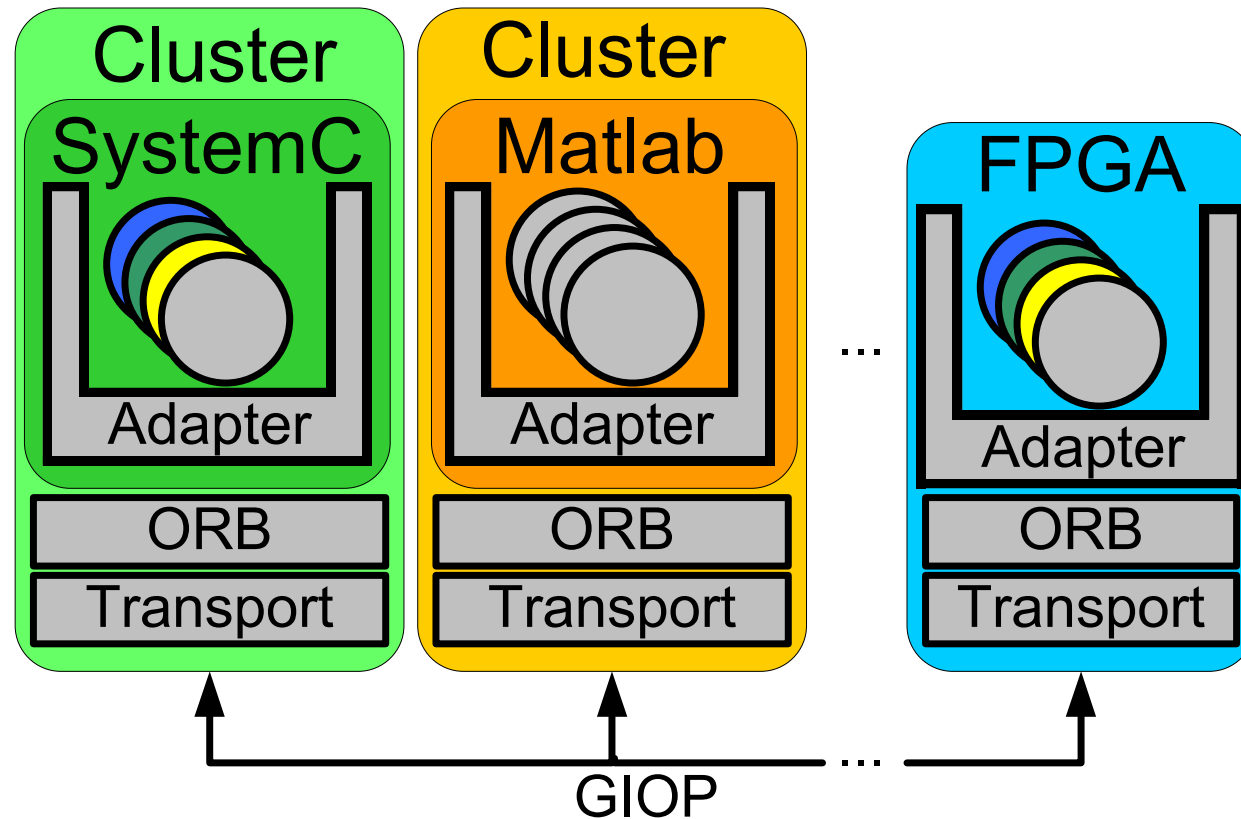
Proposed Framework: First phase

- Design spreads across two locations
- Communication via ORB
- FPGA hardware
- Multiple languages and abstraction levels



Proposed Framework: 2nd phase

- Deploy on multiple nodes
- Broader tool base



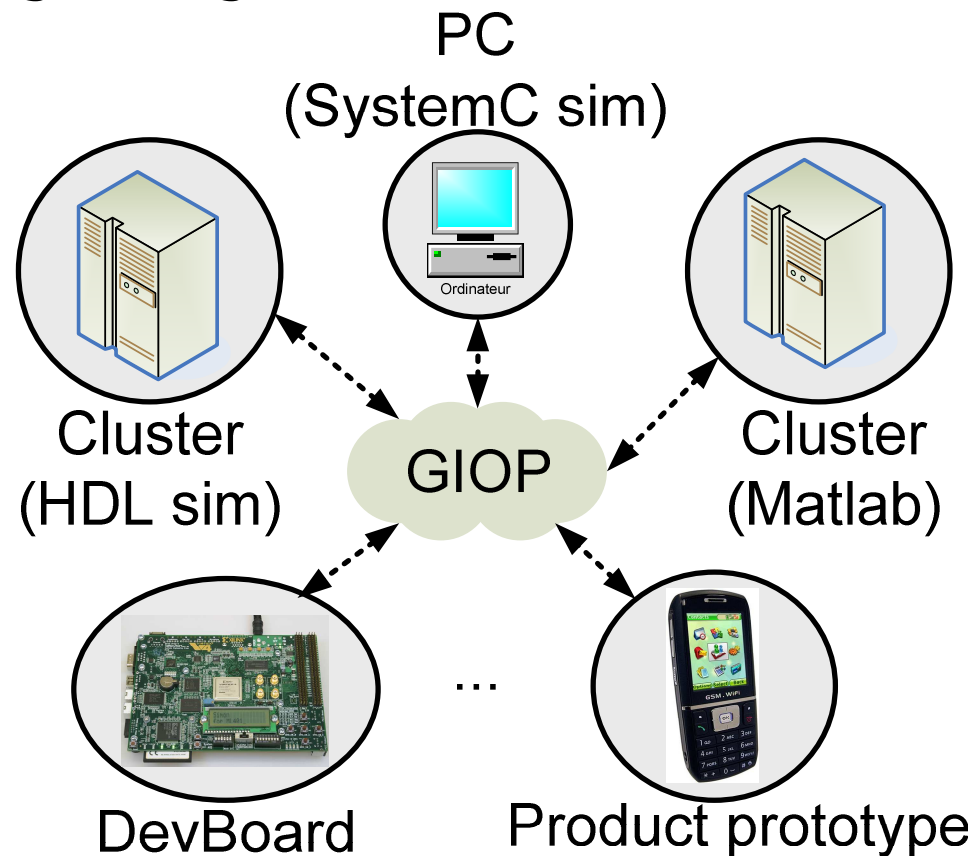


Outline

- Problems
- Proposed Methodology
- Proposed Framework
- ***Open issues***
- Conclusions and Future Work

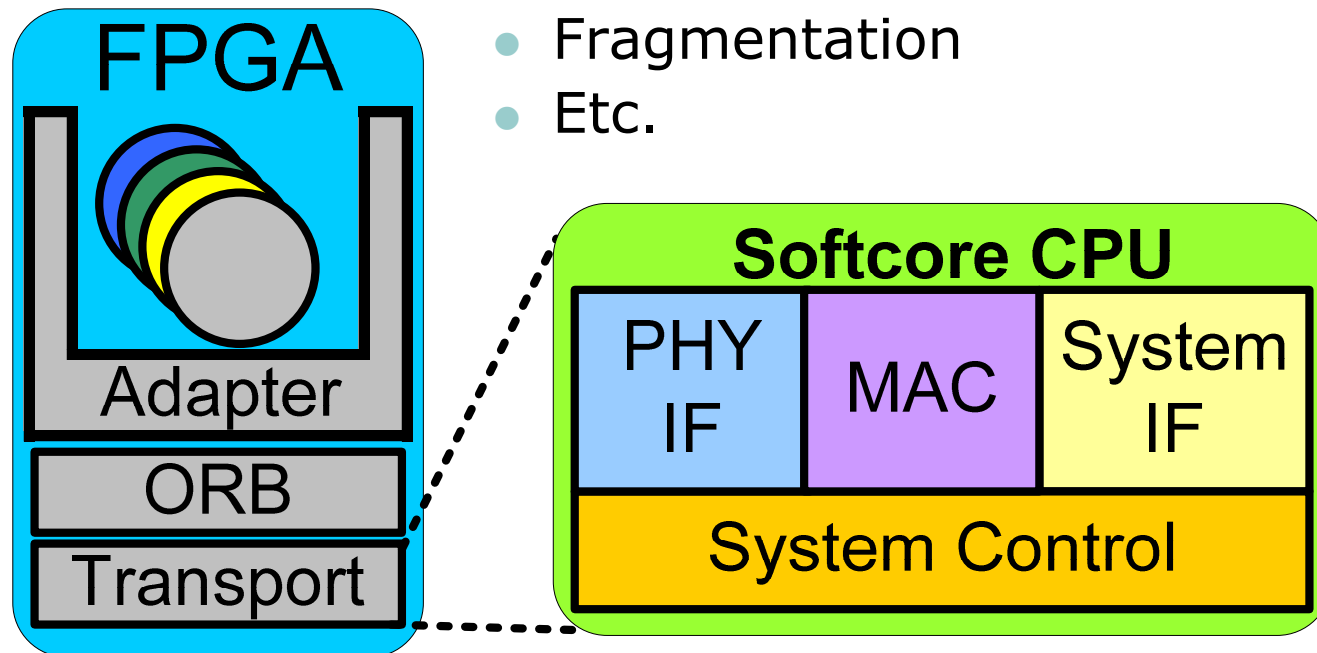
Open issues: Part 1

- Large deployment
 - E.g. Using a Cluster for HDL simulation



Open issues: Part 2

- Communication performance
 - Latency
 - Throughput
 - Fragmentation
 - Etc.





Outline

- Problems
- Proposed Methodology
- Proposed Framework
- Open issues
- ***Conclusions and Future Work***



Conclusion: Part 1

- Distributed object architecture verification
 - Hardware In the Loop
 - Cluster farms
 - Etc.
- Promotes open standards
- Promotes code reuse
- Promotes early hardware verification



Conclusion: Part 2

- Seamless integration of multiple:
 - levels of abstraction
 - design languages
 - “*physical*” location
- Allows progressive refinements towards target platform
- Provides an expendable framework
- Supports traditional verif. flow



Future Work

- Short term:
 - Complete implementation
 - Other application areas
- Mid term:
 - Performance evaluation
 - Support more 3rd party tools
- Long term:
 - Integration with the GreenSocs project



Questions?

Thank you for listening!

Contact me at

Pascal.Giard.1@ens.etsmtl.ca



Hidden slides



Related works

- VirginiaTech's CARH
 - Service-Oriented Architecture for Validating System-Level designs
 - Integrates CORBA to OSCI SystemC
 - Requires modifications to OSCI SystemC compiler
 - Not meant for hardware component interoperability ⇒ different scope