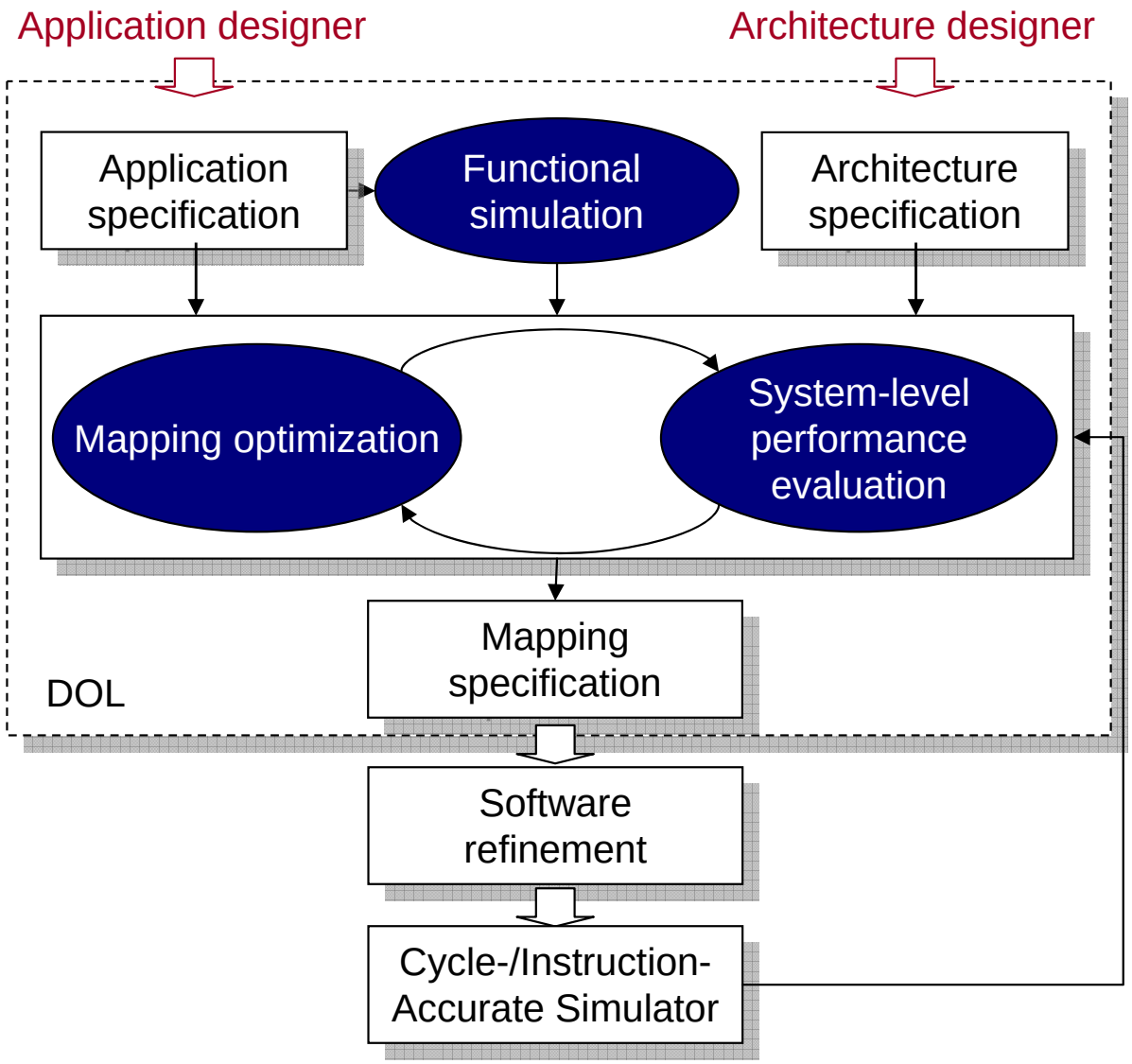
Distributed Operation Layer

Iuliana Bacivarov, Wolfgang Haid,
Kai Huang, and Lothar Thiele
ETH Zürich

Outline

- Distributed Operation Layer Overview
  - Specification
    - Application
    - Architecture
    - Mapping
  - Design Space Exploration
    - Mapping Optimization
    - Performance Evaluation
- Demo

Distributed Operation Layer Overview



Context

- **Highly parallel applications**
  - e.g. wave field synthesis, physical particles modeling, ultrasound scanners
- **Multi-processor architectures**
  - tile-based platform

DOL Framework

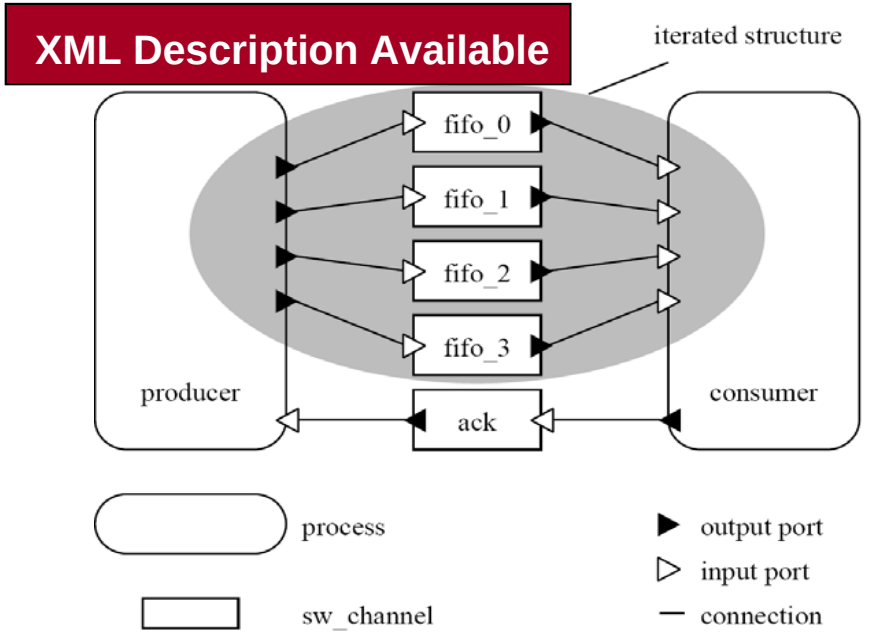
- Scalable manner to specify applications and architectures
- Functional simulation for debugging and profiling
- Design space exploration: performance analysis and optimization

Application Specification

Structure

- **Process Network**
  - Processes
  - SW channels (FIFO behavior)
- **Iterators**
  - Scalability for processes, SW channels, entire structures
- **Annotations**
  - e.g. channel size, process runtimes

XML Description Available



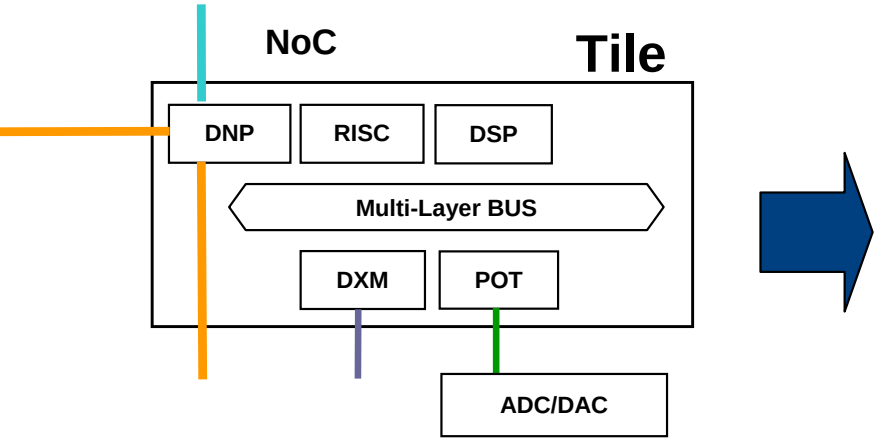
Functional specification

- **Language:** C/C++
- **Specific programming DOL API**

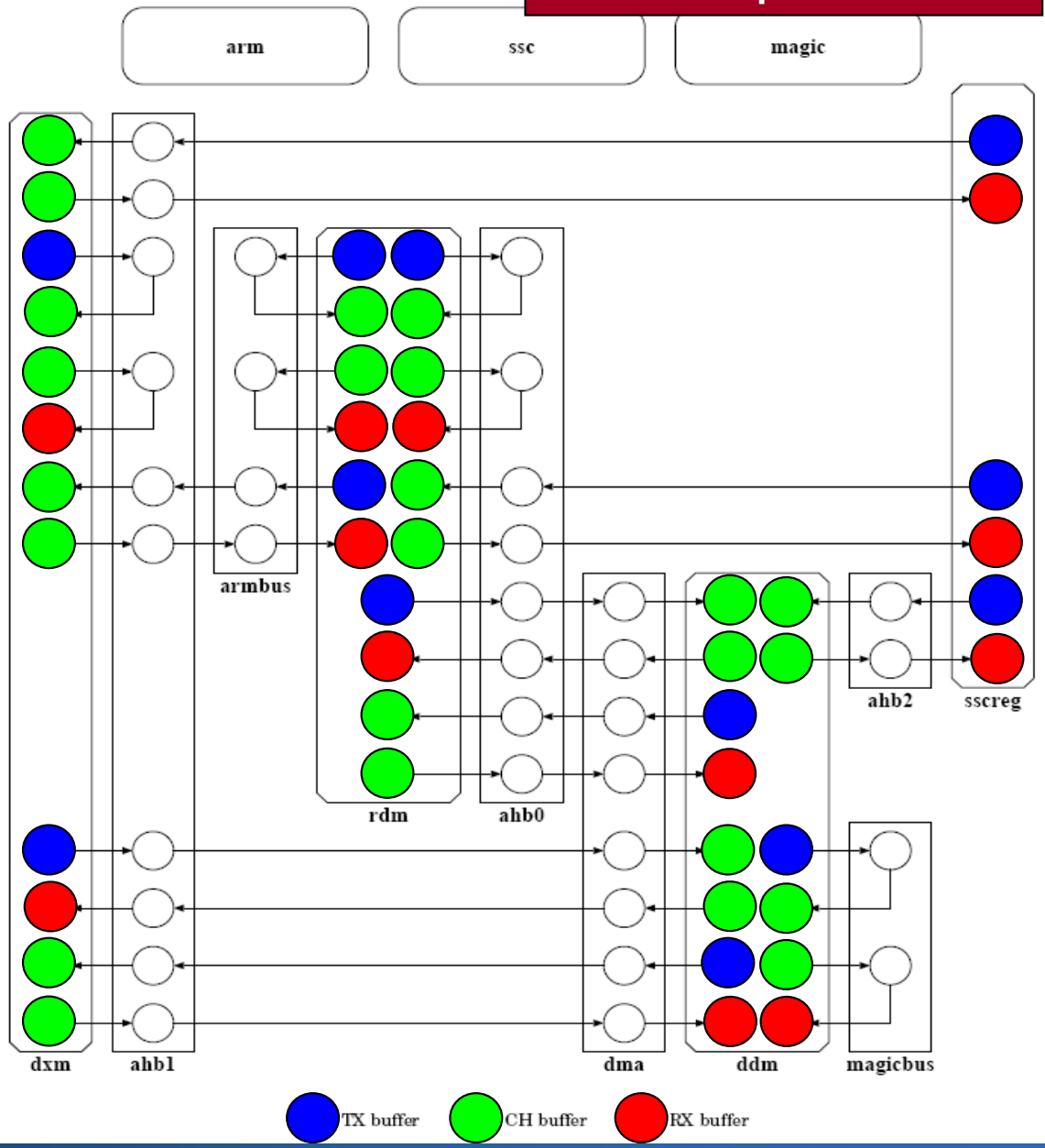
Algorithm 1 Process Model

```
1: procedure INIT(DOLProcess p)                ▷ initialization
2:     initialize local data structures
3: end procedure

4: procedure FIRE(DOLProcess p)                ▷ execution
5:     DOL_read(INPUT, size, buf)              ▷ blocking read
6:     manipulate
7:     DOL_write(OUTPUT, size, buf)            ▷ blocking write
8: end procedure
```

Architecture Specification



XML Description Available



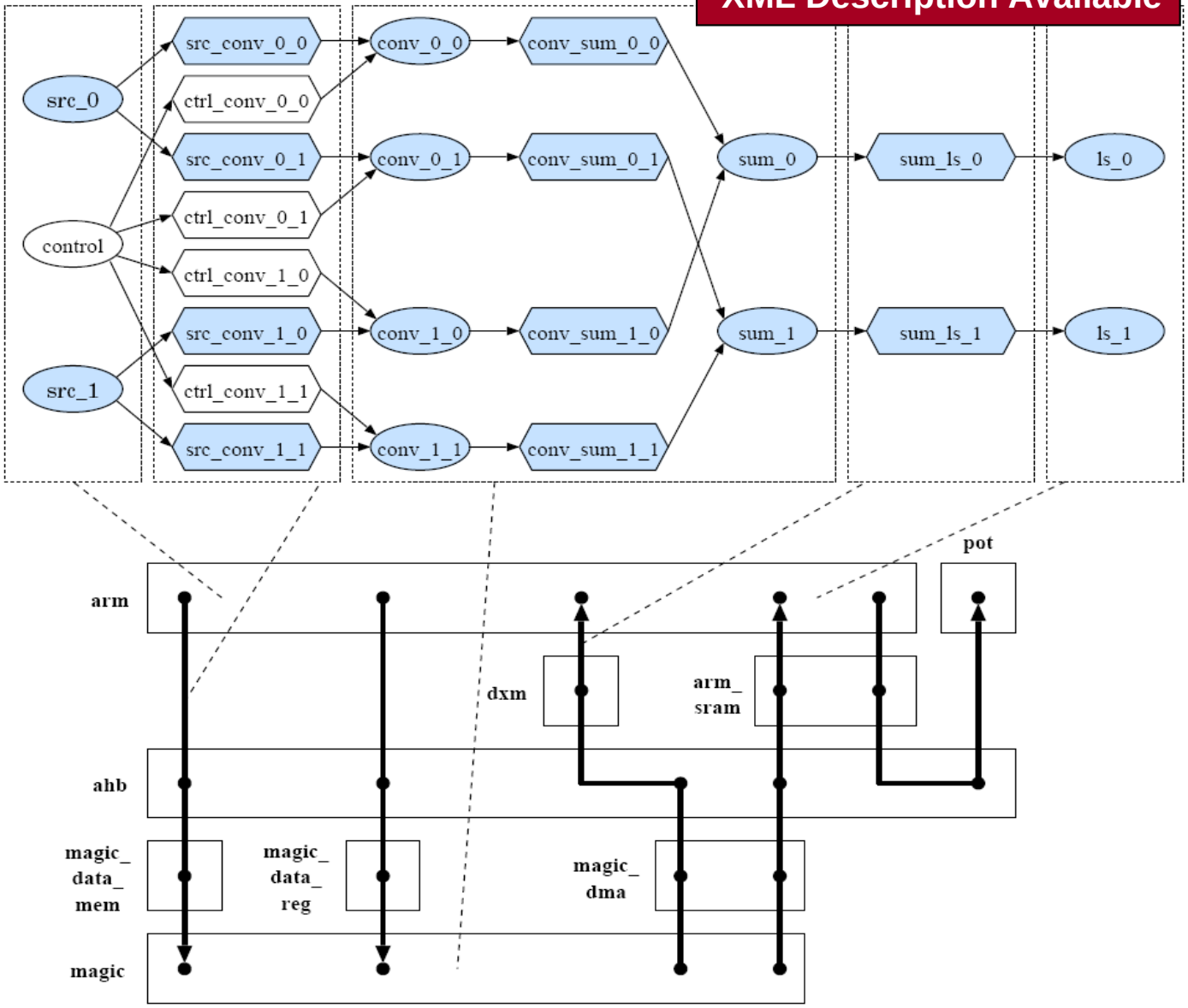
- **Abstract platform modeling**
  - Processors, peripherals, memories, buses, etc.
  - Read and write communication paths
  - Performance data:

    latency and bandwidth of HW communication channels

Mapping Specification

- **Binding**
  - Processes to execution resources
  - SW channels to read/write paths
- **Scheduling**
  - Processors
  - Communication
- **Constraints**
  - To be considered during mapping

XML Description Available

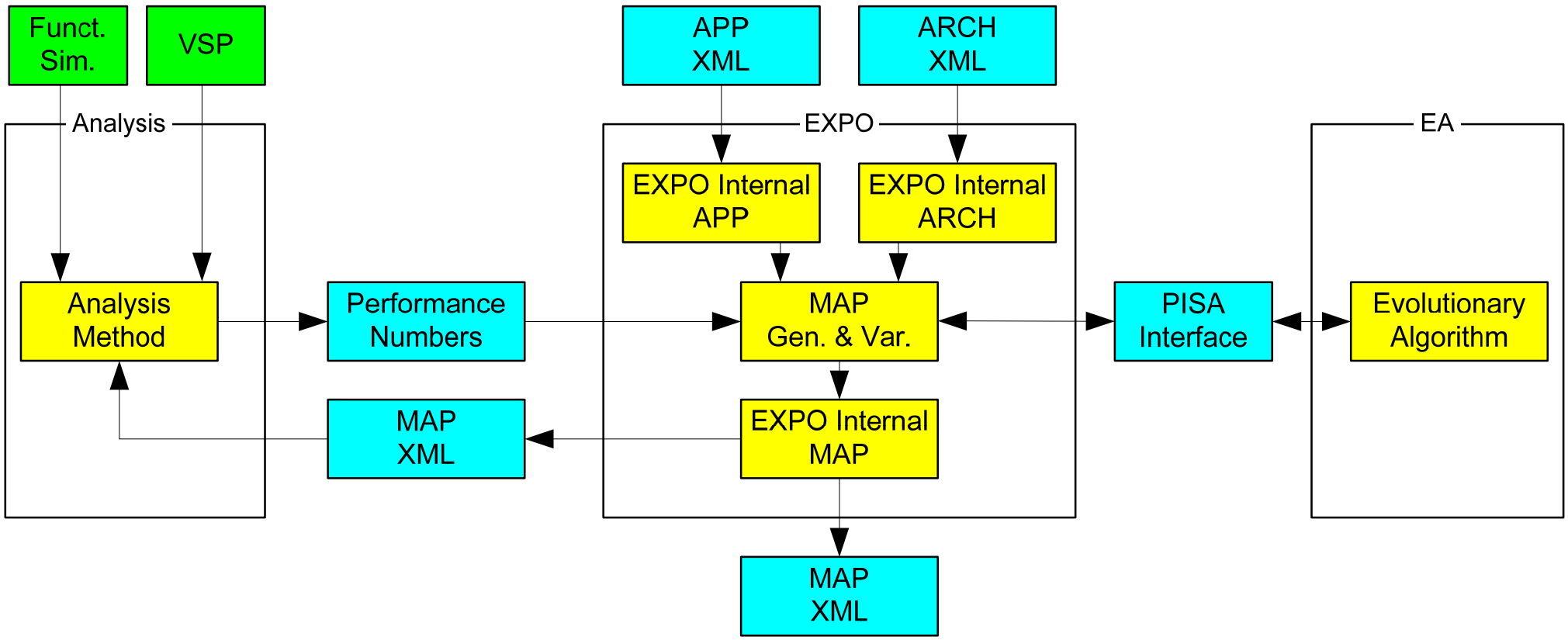


Outline

- Distributed Operation Layer Overview
  - Specification
    - Application
    - Architecture
    - Mapping
  - **Design Space Exploration**
    - **Mapping Optimization**
    - **Performance Evaluation**
- Demo

Design Space Exploration Framework

- Modular framework - **optimization algorithms** and **performance analysis** plug-ins
- PISA&EXPO: multi-objective optimization using evolutionary algorithms
  - [PISA] https://www.tik.ee.ethz.ch/pisa; [EXPO] https://www.tik.ee.ethz.ch/expo



Additive Model for Performance Analysis

- **Processor execution time:**

$$T(\text{processor } c) = \sum_{\forall p \text{ mapped onto } c} r(p,c) \qquad (1)$$

  - *r(p; c)* - runtime of the *process p* on the *processor c* → **VSP simulation**

- **Transfer time of a communication resource:**

$$T(\text{HW channel } g) = \sum_{\forall s \text{ mapped onto } g} d(s)/b(g) \qquad (2)$$

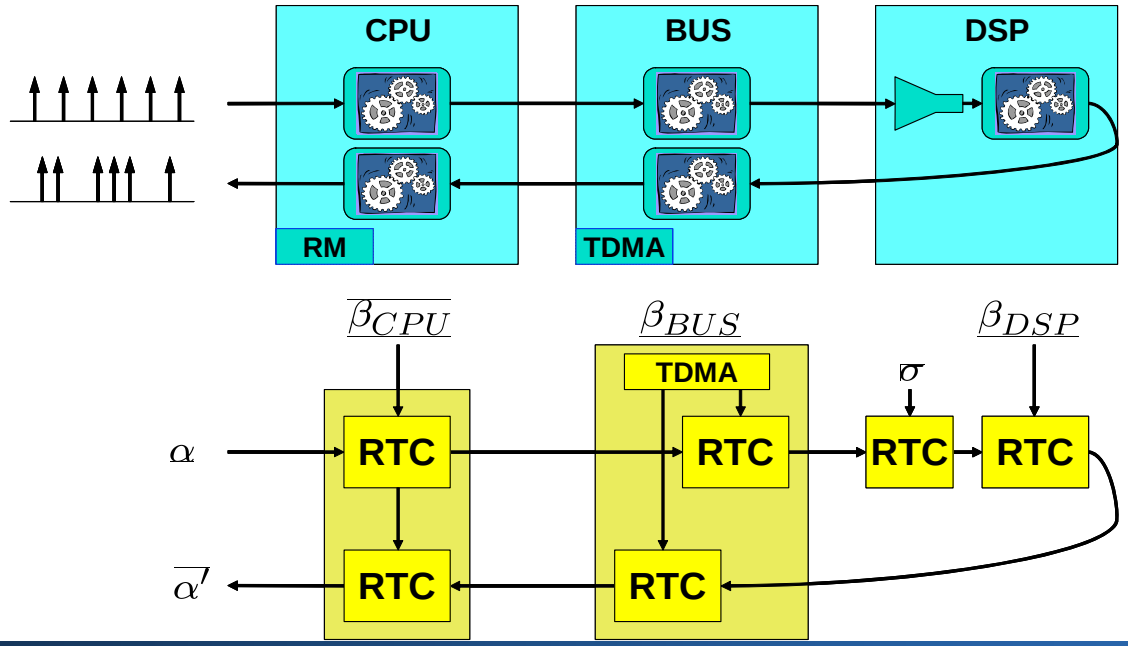
  - *d(s)* - nb. of bytes comm. over *SW channel s* → **Functional simulation**
  - *b(g)* - bandwidth of the *resource g* → **Platform benchmarking**

Simplified, static model

- ***Parameters:*** resource allocation, binding of processes and SW channels to resources
- ***Does not account:*** computation and communication overhead, blocking times, HW channels usage should be below a certain threshold (up to 60% of the max bandwidth)

On-Going & Future Work in Performance Analysis

- **Objective:** more accurate and complex performance analysis model, i.e. including dynamic features
- **MPA – Modular Performance Analysis** (http://www.mpa.ethz.ch/)
  - Modular system composition; Abstract event processing unit
  - *Dynamic aspects included: scheduling*
- **On-going research:**
  1. **Complex activation schemes:** ref: W. Haid, L. Thiele, "Complex Task Activation Schemes in System Level Performance Analysis", CODES+ISSS'07
  2. **Timing correlation:** ref: K. Huang, L. Thiele, T. Stefanov, E. Deprettere, "Performance Analysis of Multimedia Applications using Correlated Streams", DATE'07

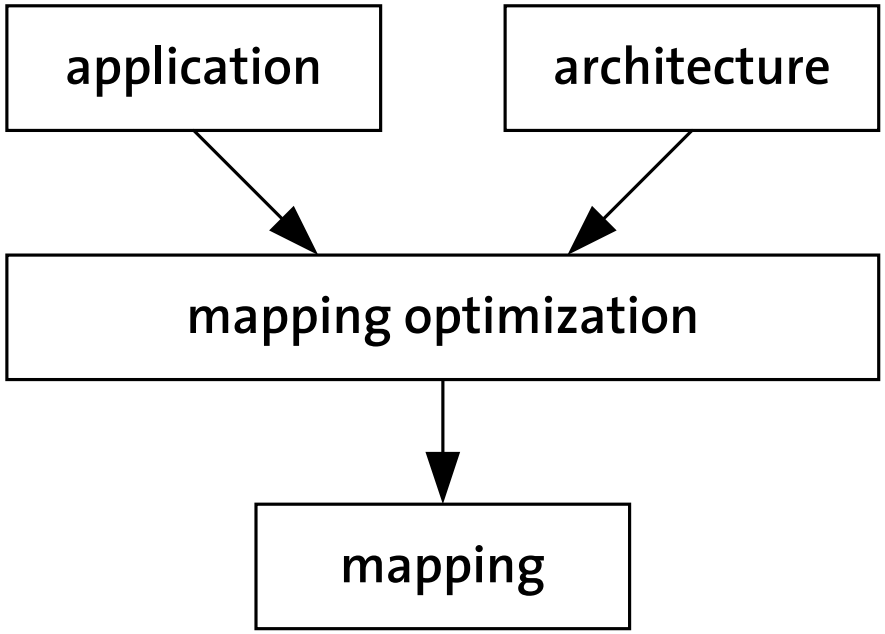


Outline

- Distributed Operation Layer Overview
  - Specification
    - Application
    - Architecture
    - Mapping
  - Design Space Exploration
    - Mapping Optimization
    - Performance Evaluation
- **Demo**

Objective and Design Trajectory

- **Objective:** map a parallel application onto a parallel, heterogeneous architecture



Design Trajectory

- **System specification**
  - Application
  - Architecture
- **Application profiling**
  - Functional simulation
  - Instruction-/cycle-accurate simulation
- **Mapping optimization**
  - Performance estimation using analytic model
  - Optimization algorithm

Design Trajectory

application developer

process network XML

C source code

distributed operation layer

XML visualization

functional simulation

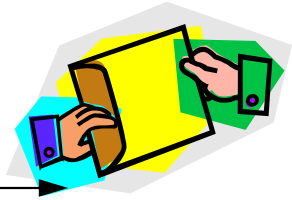
instr.-accurate simulation

ann. process network XML

architecture XML

mapping design space exploration

mapping XML



to HdS refinement

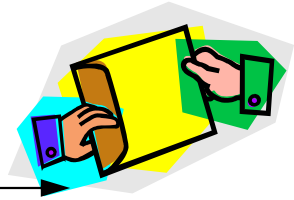
Design Trajectory: Specification

application developer

XML

DOTTY

XML



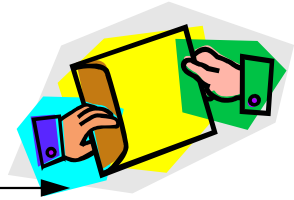
to HdS refinement

Design Trajectory: Functional Simulation

application developer

~/shapes/dolPrototype/trunk/build/bin/main

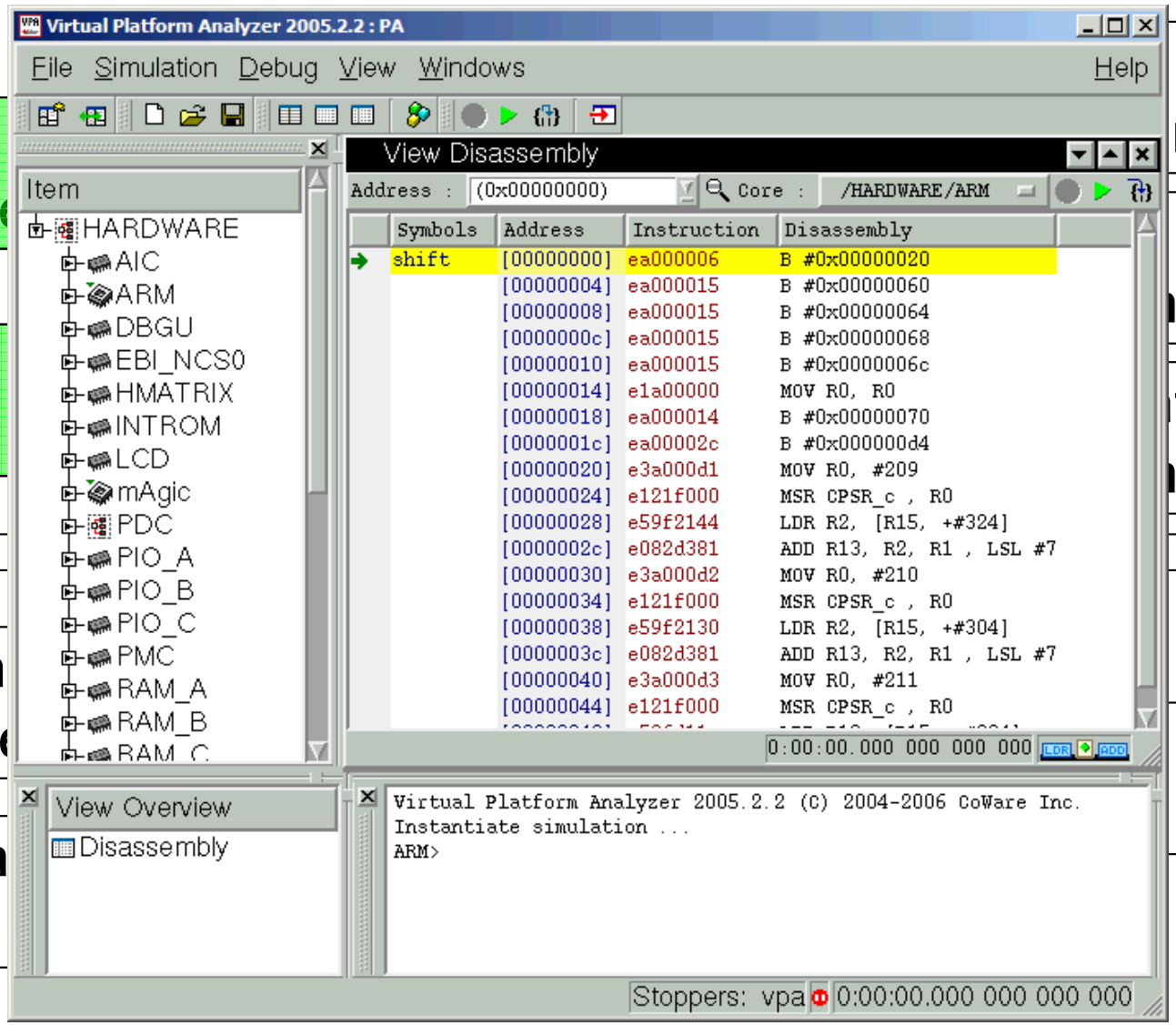
```
$ ./wfs/systemc/src/sc_application.exe

         SystemC 2.1.v1 --- Aug  3 2006 18:42:51
     Copyright (c) 1996-2005 by all Contributors
                 ALL RIGHTS RESERVED
src_0 = [];
src_0 = [src_0; 0.000000];
src_0 = [src_0; 0.000000];
src_0 = [src_0; 0.039260];
src_0 = [src_0; 0.078459];
src_0 = [src_0; 0.117537];
src_0 = [src_0; 0.156434];
src_0 = [src_0; 0.195090];
src_0 = [src_0; 0.233445];
src_0 = [src_0; 0.271440];
src_0 = [src_0; 0.309017];
src_0 = [src_0; 0.346117];
src_0 = [src_0; 0.382683];
src_0 = [src_0; 0.418660];
src_0 = [src_0; 0.453991];
src_0 = [src_0; 0.488621];
src_0 = [src_0; 0.522499];
src_0 = [src_0; 0.555570];
src_0 = [src_0; 0.587785];
src_0 = [src_0; 0.619094];
src_0 = [src_0; 0.649448];
src_0 = [src_0; 0.678801];
src_0 = [src_0; 0.707107];
src_0 = [src_0; 0.734323];
src_0 = [src_0; 0.760406];
src_0 = [src_0; 0.785317];
src_0 = [src_0; 0.809017];
src_0 = [src_0; 0.831470];
src_0 = [src_0; 0.852640];
src_0 = [src_0; 0.872496];
src_0 = [src_0; 0.891007];
src_0 = [src_0; 0.908143];
src_0 = [src_0; 0.923880];
src_1 = [];
src_1 = [src_1; 0.000000];
src_1 = [src_1; 0.000000];
src_1 = [src_1; 0.078459];
src_1 = [src_1; 0.156434];
src_1 = [src_1; 0.233445];
src_1 = [src_1; 0.309017];
```

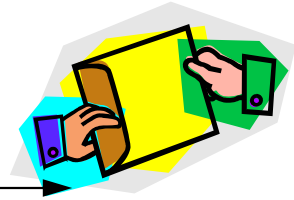


to HdS refinement

Design Trajectory: Virtual Simulation Platform

application developer





to HdS refinement

Design Trajectory: Profiling

application developer

process network XML

C source code

distributed operation layer

XML visualization

functional simulation

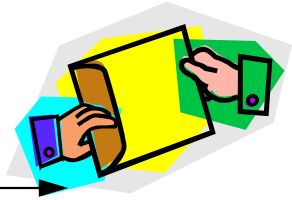
instr.-accurate simulation

ann. process network XML

architecture XML

mapping design space exploration

mapping XML



to HdS refinement

Design Trajectory: Optimization

application developer

net

C sc

XML

ualization

unctional

mulation

rate

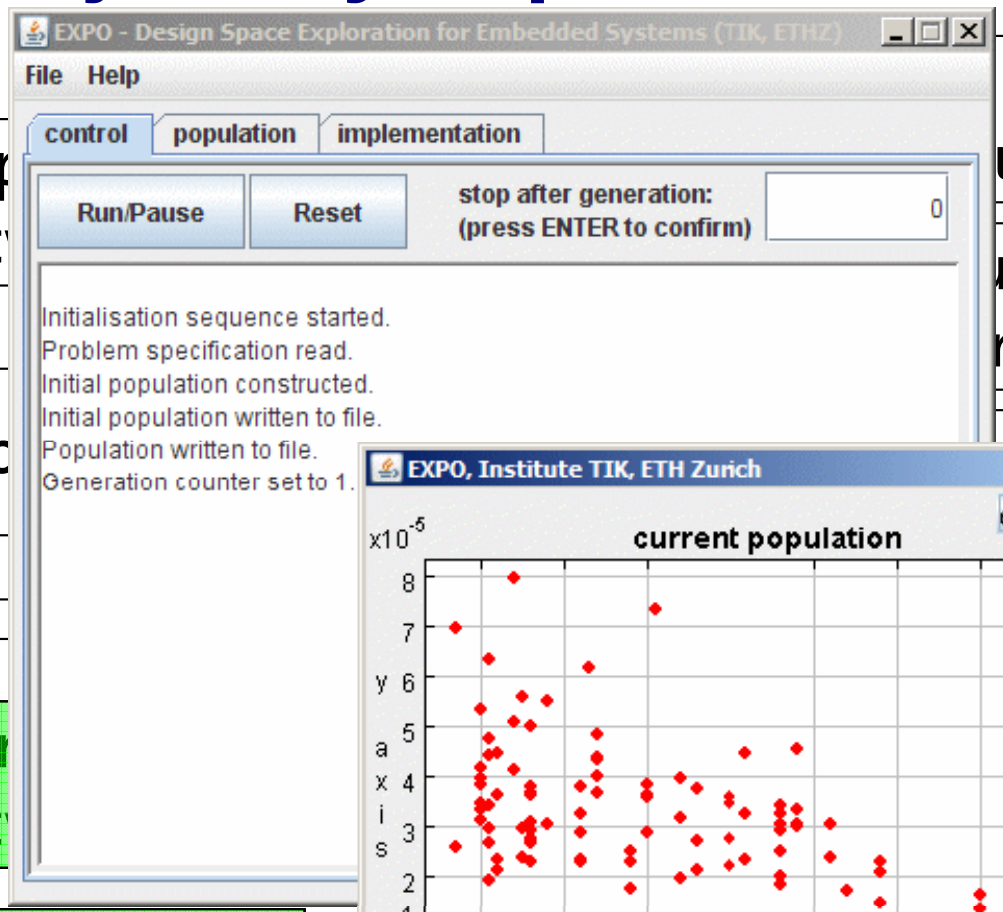
on

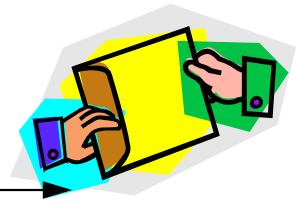
ann

net

g

architecture XML

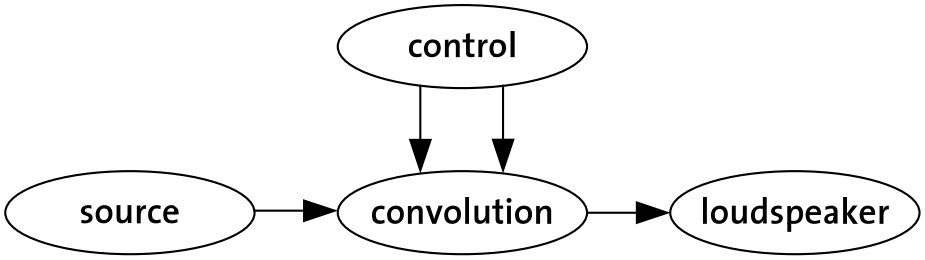




to HdS refinement

System Specification: WFS Application

- **Structure**
  - ***XML representation*** of the Kahn process network for Wave Field Synthesis
- **Functionality**
  - ***ANSI C & DOL API***

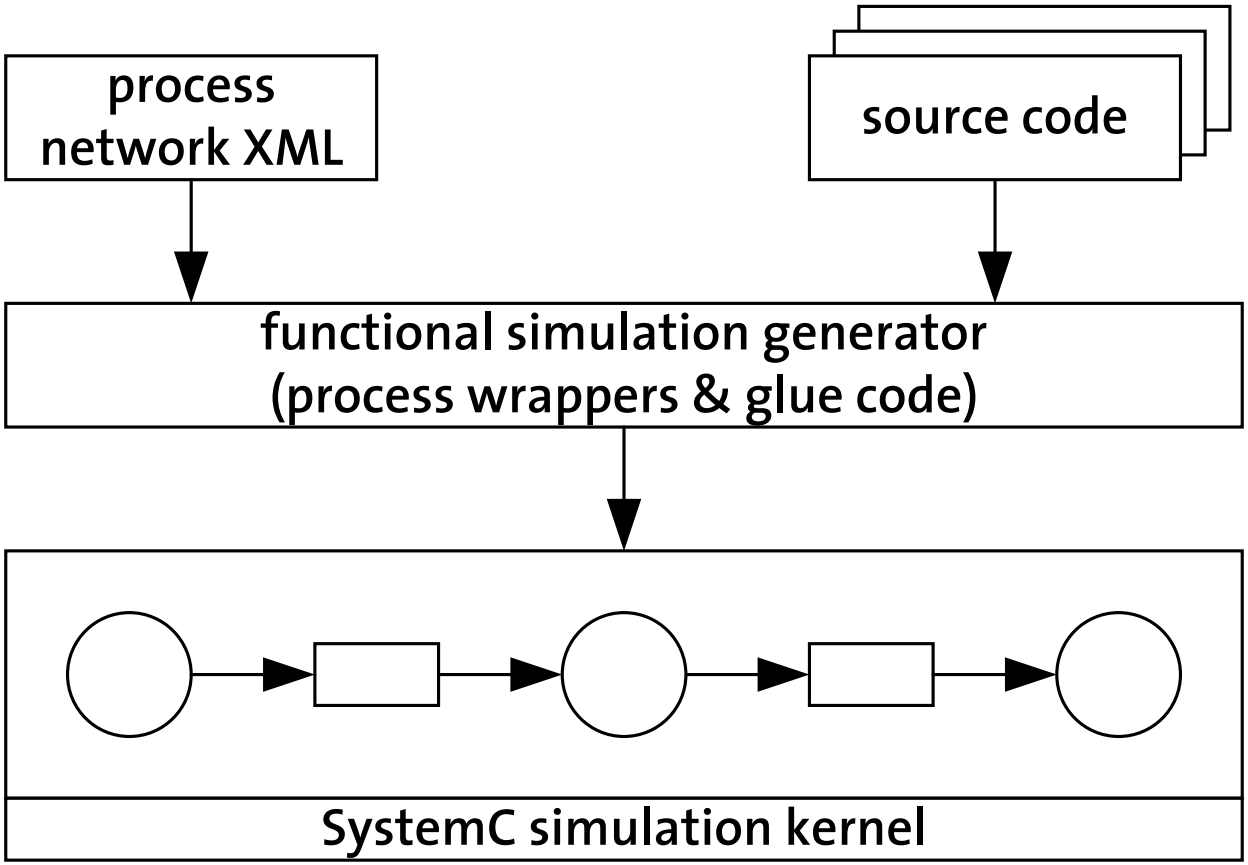


Algorithm 1 Process Model

```
1: procedure INIT(DOLProcess p)                ▷ initialization
2:     initialize local data structures
3: end procedure

4: procedure FIRE(DOLProcess p)                ▷ execution
5:     DOL_read(INPUT, size, buf)              ▷ blocking read
6:     manipulate
7:     DOL_write(OUTPUT, size, buf)            ▷ blocking write
8: end procedure
```

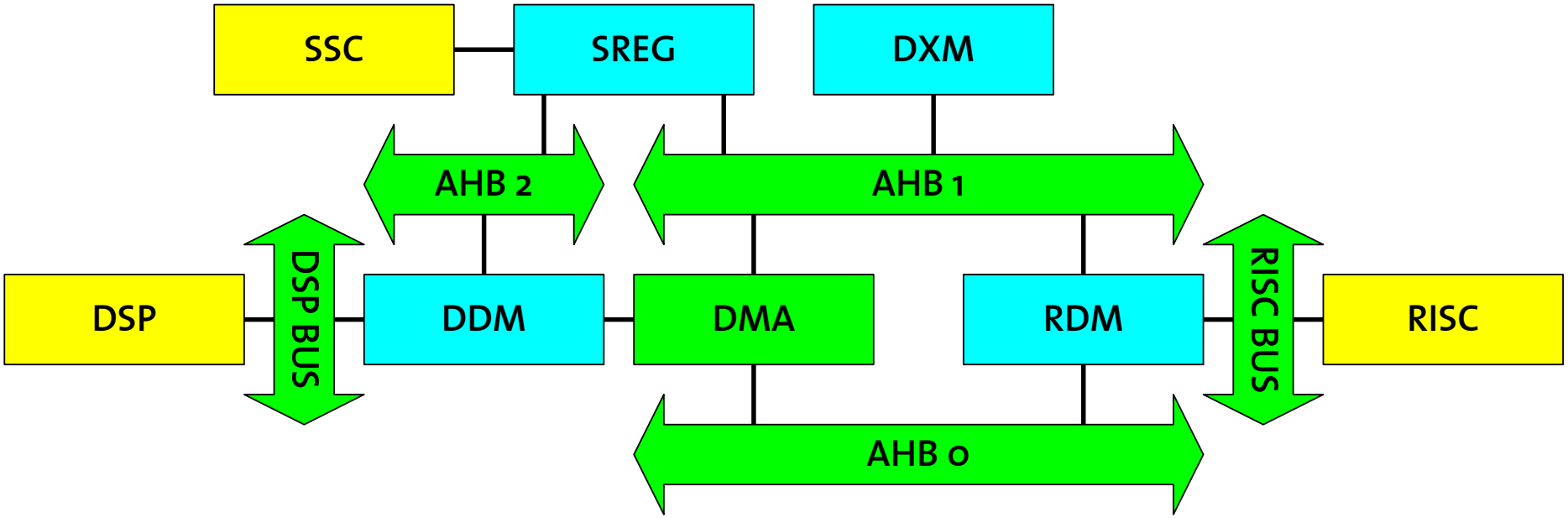
Functional Simulation

- The ***functional simulation*** is a tool to develop, debug, and test applications



System Specification: Architecture

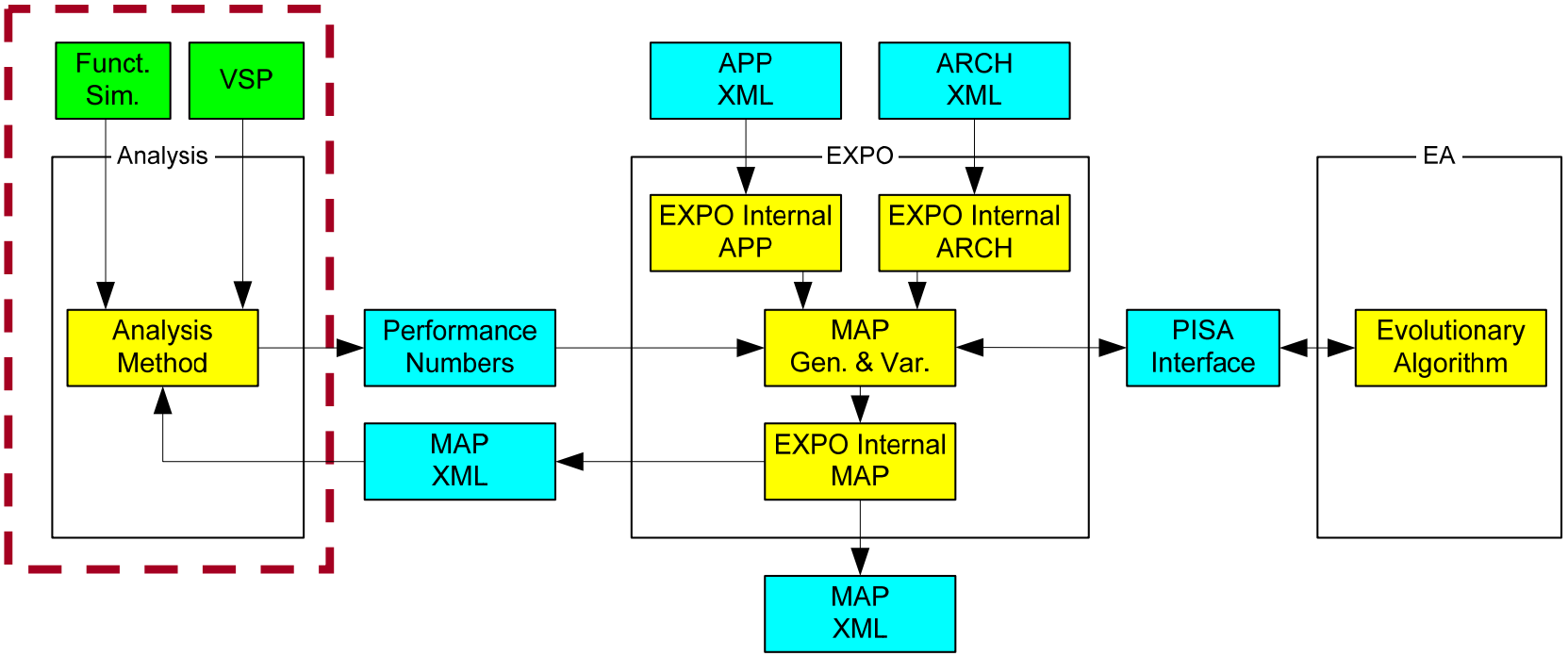
- Architecture modeled at an abstract level in ***XML format***
- Modeled elements:
  - Processors, buses, memories
  - Communication paths between these elements



Mapping Optimization Framework

Approach:

- Profile application (*once*) using functional and instruction-accurate simulation
- Perform mapping optimization using an analytic model



Where are data obtained from?

processor c with worst total runtime

number of activations of process p

runtime of process p on processor c

$$obj_1 = \max_{c \in \mathcal{C}} \left\{ \sum_{\forall p \text{ mapped to } c} n(p) \cdot r(p,c) \right\}$$

communication link with worst load

communication request from channel s

bandwidth of communication link g

$$obj_2 = \max_{g \in \mathcal{G}} \left\{ \sum_{\forall s \text{ mapped onto } g} \frac{b(s)}{t(g)} \right\}$$

Static parameters:

- bus bandwidth *t(g)*

Functional simulation:

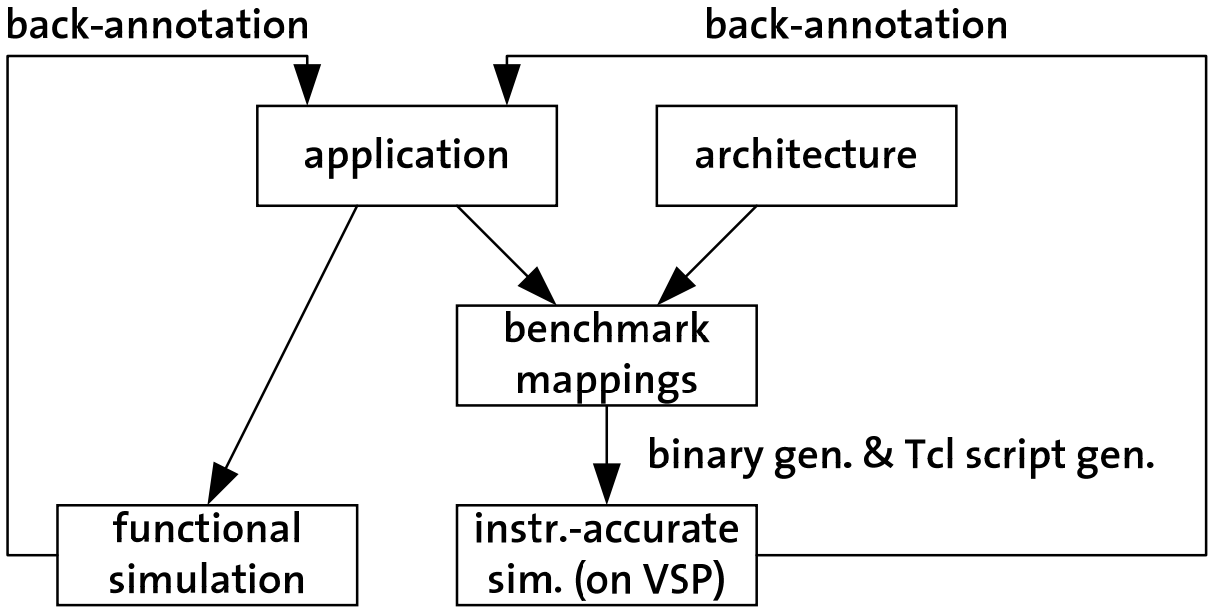
- number of activations for each process *n(p)*
- amount of data for each channel *b(s)*

VSP:

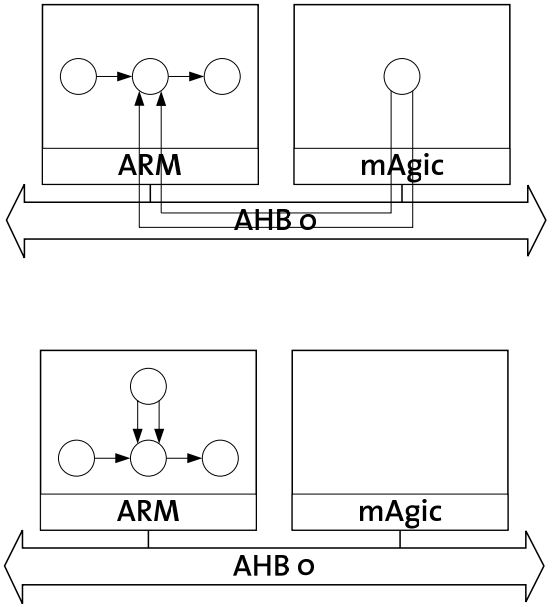
- runtime of each process on different processors *r(p, c)* (by using benchmark mappings)

Profiling VSP Execution

1. Benchmark mapping generation
2. Binary generation (using HdS & compiler), Tcl script generation
3. Execution on VSP using Tcl script (log file generation)
4. Log file analysis and back-annotation

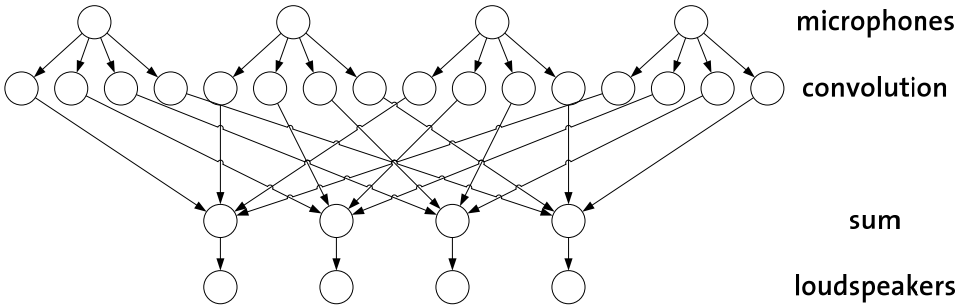


Benchmark mappings:

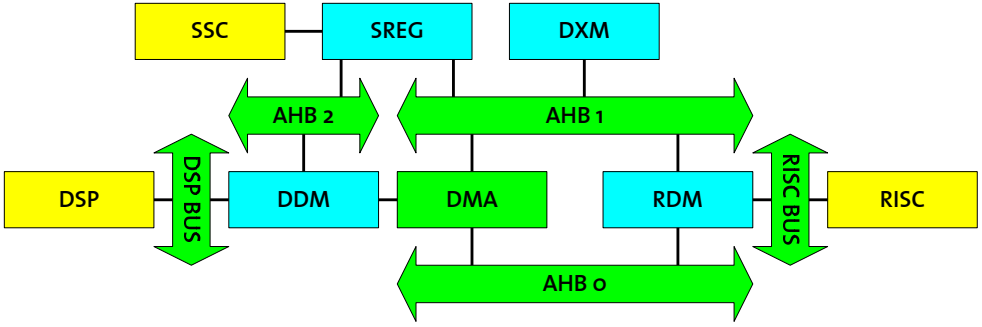


Mapping of WFS onto SHAPES Tile

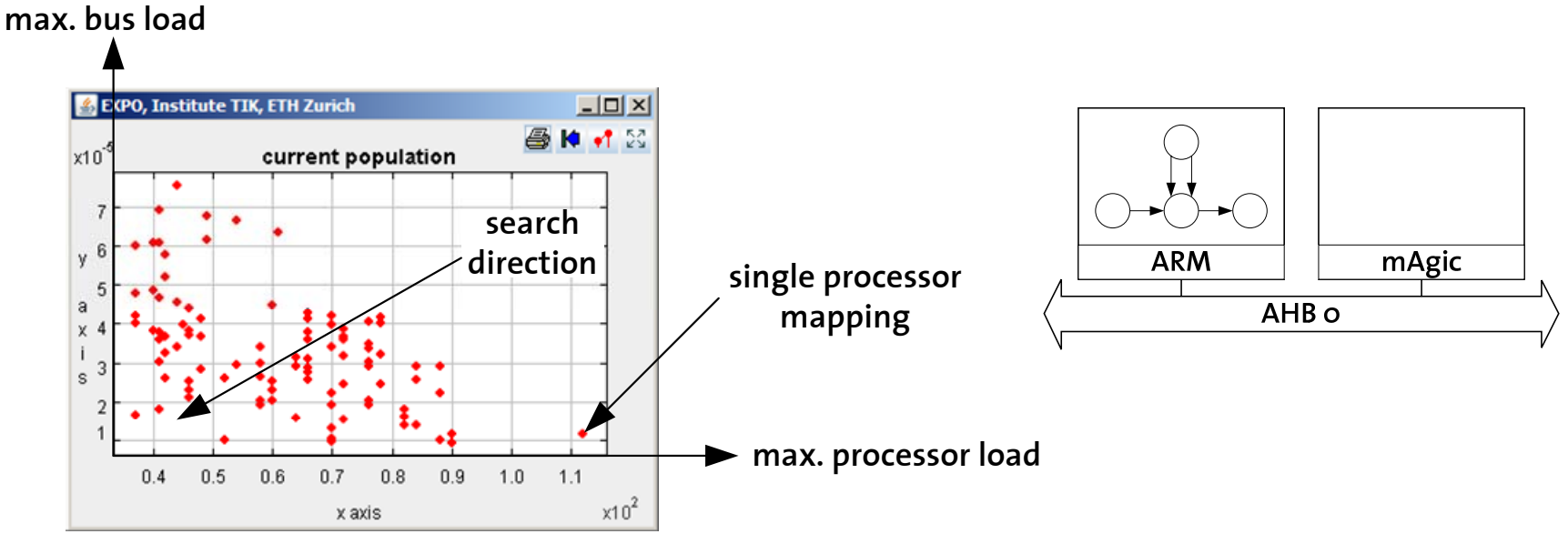
■ Process network of WFS application



■ Block diagram of SHAPES tile



■ Expo mapping example



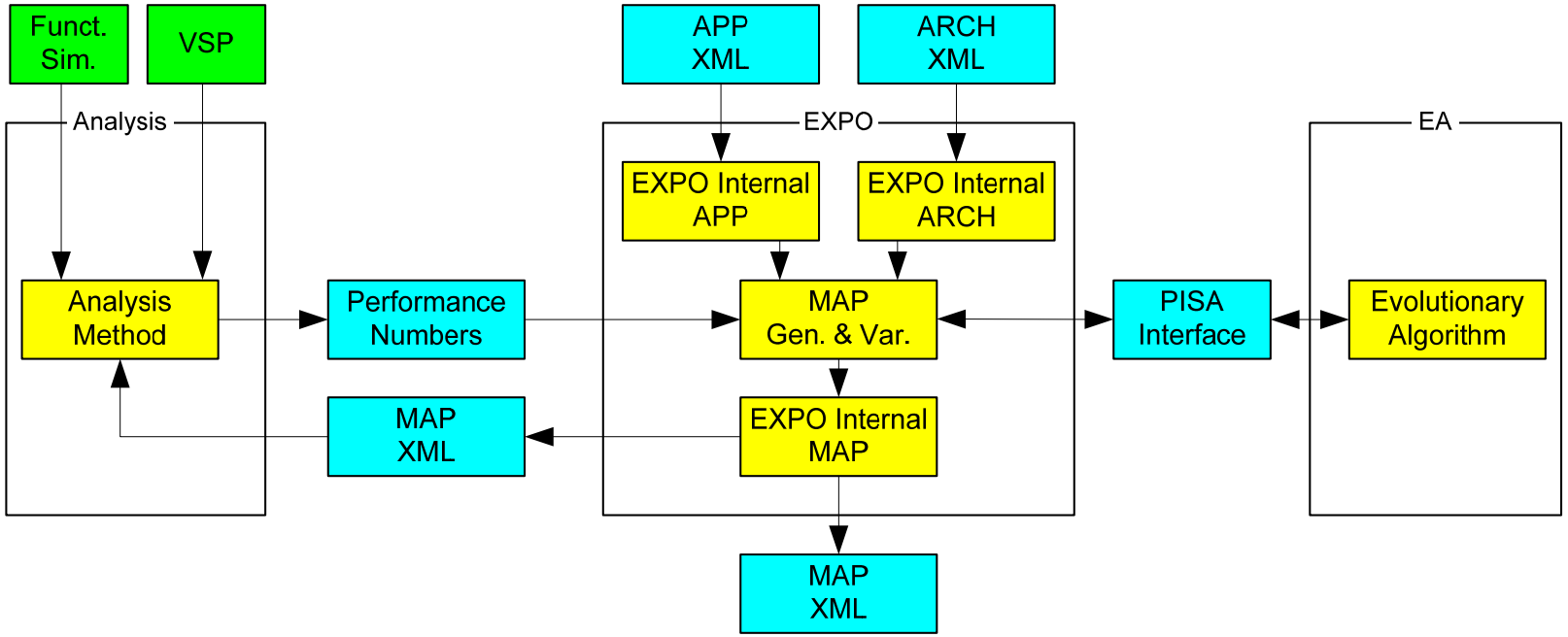
Mapping Optimization Framework

Approach:

- Profile application (*once*) using functional and instruction-accurate simulation
- Perform mapping optimization using an analytic model

Optimization tools:

- ***SPEA2***: evolutionary algorithm for optimization
- ***EXPO***
  - Mapping generation
  - Performance estimation
  - Graphical user interface



Summary

- Demonstration how a parallel application is mapped to a heterogeneous and distributed architecture
- Demonstration of integration of complete SW tool chain
  - DOL → HdS → 2 compilers → VSP (→ DOL)
  - DOL – HdS interface: XML specification
  - DOL – VSP interface: Tcl script
- EXPO framework for mapping optimization

Thank You!

Any Questions?