SystemQ
Evaluating System-Level Design Choices Combining Queuing Networks and SystemC

12th SystemC Users Group Meeting, Lausanne, Switzerland

September 27, 2005

Sören Sonntag

Matthias Gries

Christian Sauer

Raimar Thudt

Soeren.Sonntag@infineon.com

Never stop thinking.

Never stop thinking

Outline

1. Motivation
2. Introduction to SystemQ
3. Case study
4. Results and discussion

Designer's Dilemma

- Embedded systems
  - Computational complexity ↑
  - Novel parallel and programmable architectures ↑
  - Hardware *and* software concerns ↑
  - Time to market ↓
  - Platform costs ↓
  - Power dissipation ↓
- Need for
  - Early design decisions
  - Performance estimations even in concept phase
  - Mapping of functionality onto computing resources
  - First time right

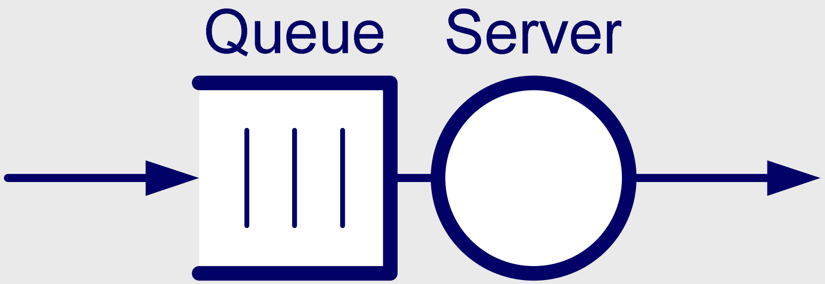
SystemC-based Simulation

- Pros
  - Discrete event simulation capabilities
  - Supports different abstraction levels
  - Refinement possible
  - Distinct modules and communication

- Cons
  - Focused on
    - Transaction level
    - RT level

Never stop thinking

Performance Evaluation

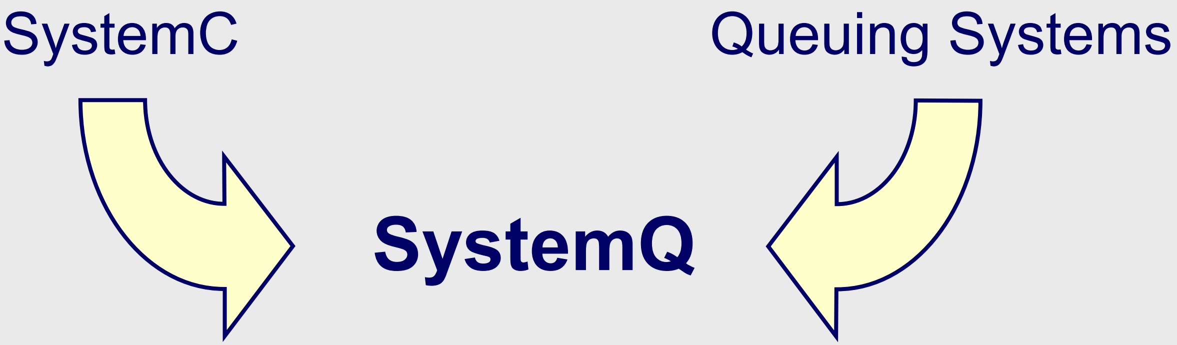


- Queuing systems
  - Important analytical modeling technique
  - Steady-state analysis
  - Explicit scheduling
- Typical measures
  - Residence time of transactions
  - Average queue length
  - Server utilization
- Cons
  - No transient analysis
  - No path to implementation

Never stop thinking

Never stop thinking

SystemQ Overview



SystemC

Queuing Systems

SystemQ

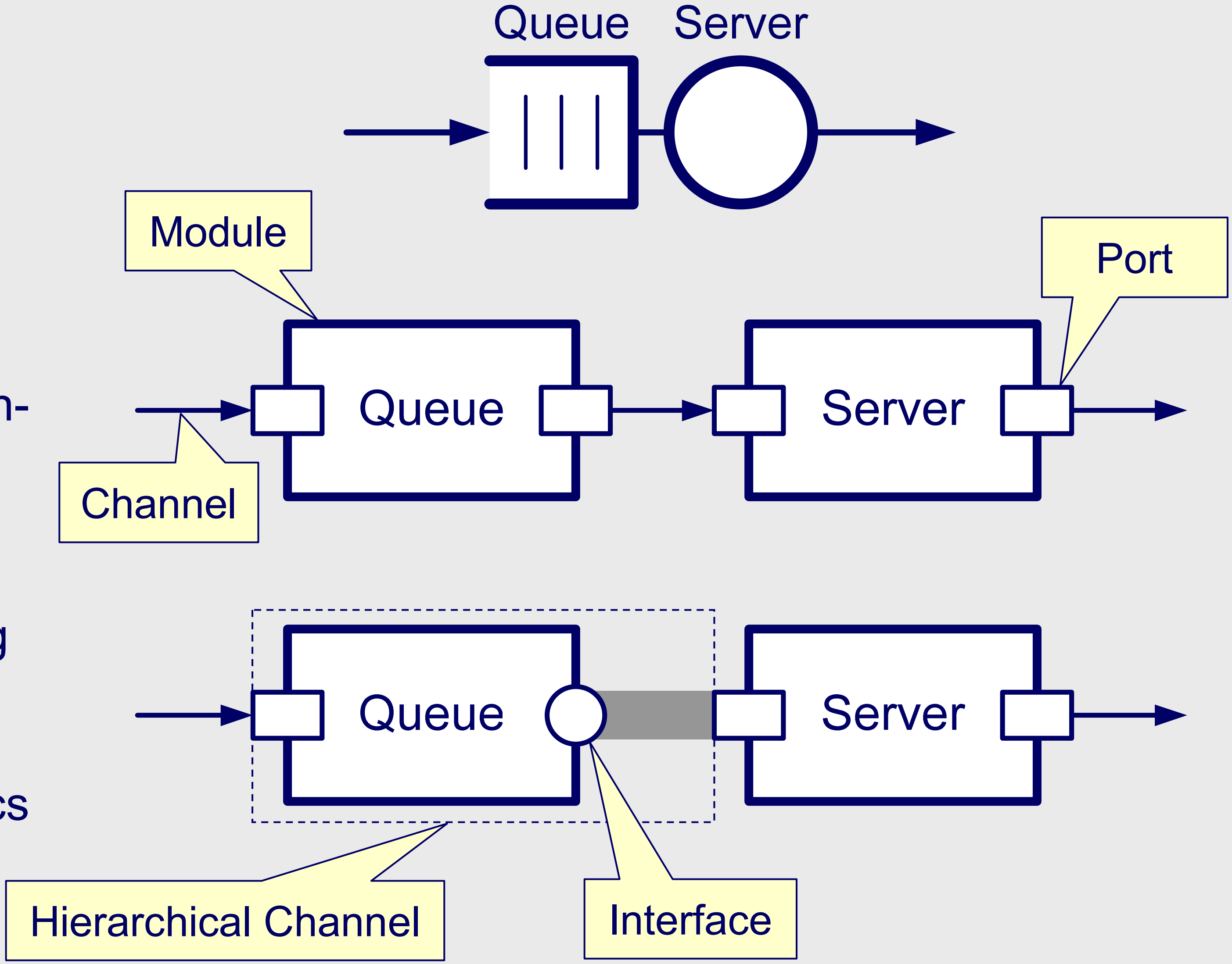
- Rich class library
- Exploitation of OO programming
- Simulation even at concept phase of design
- Workload-dependent behavior can be simulated
- Systematic refinement steps
- Detailed insight into system

SystemQ Models

Queuing Theory

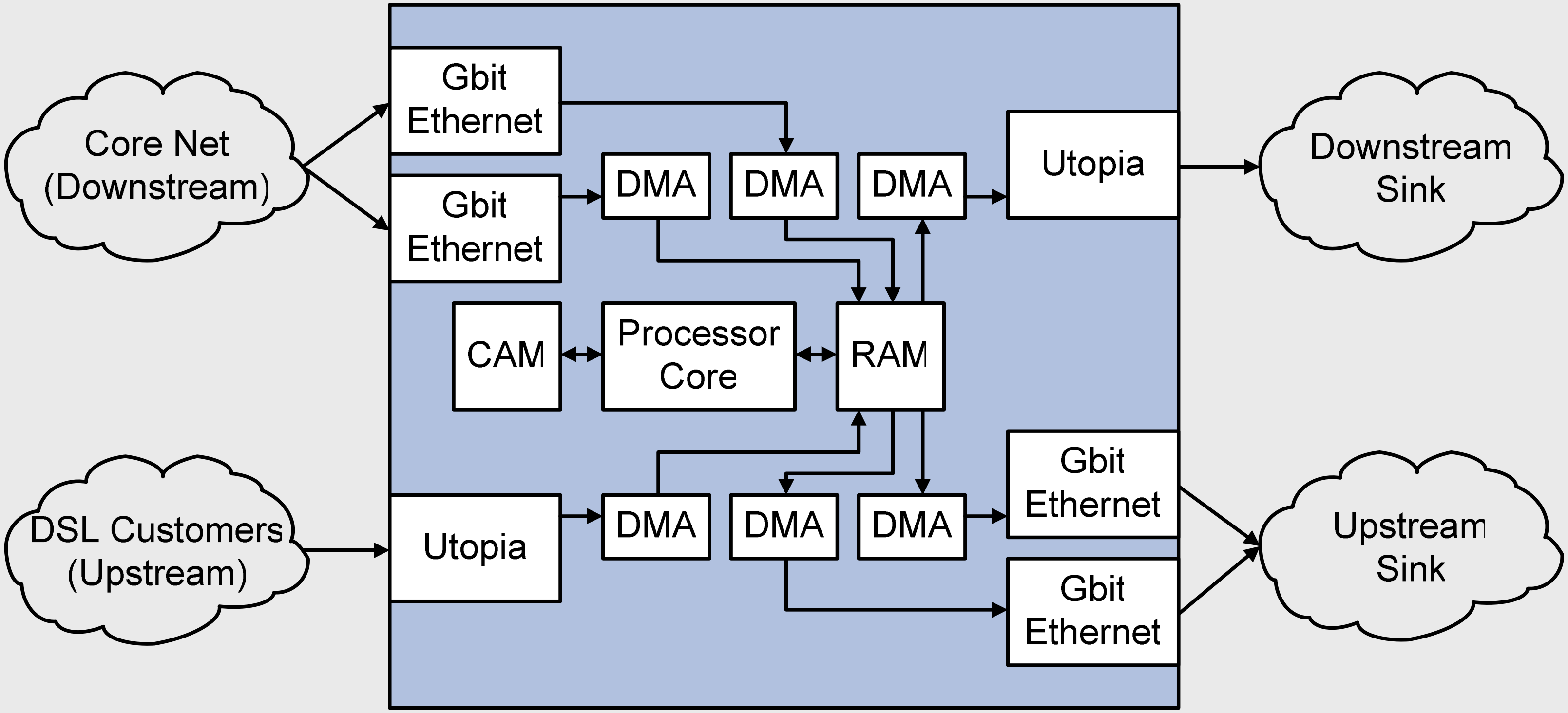
SystemC Representation

Enforcing Queuing System Semantics

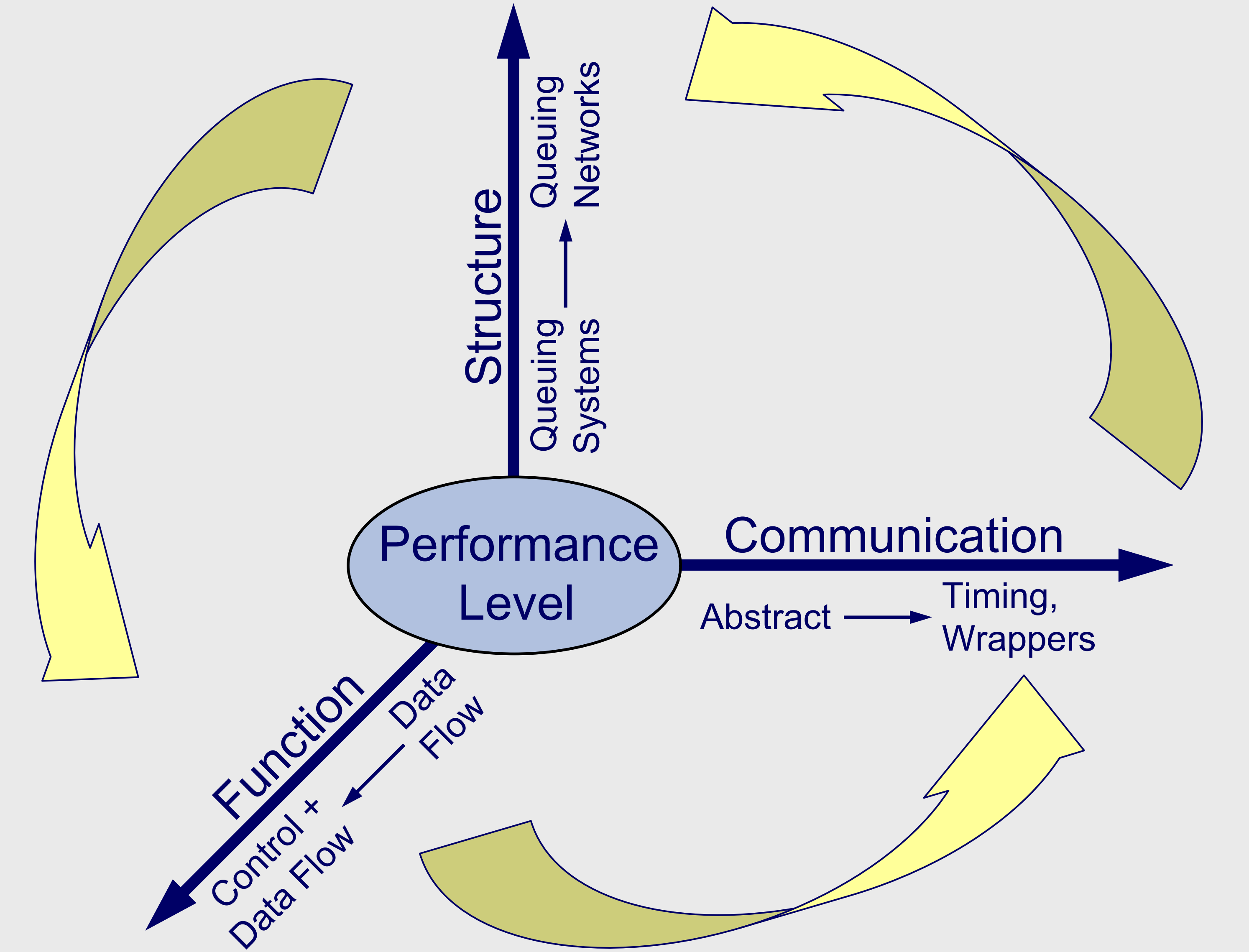


Never stop thinking

Block Diagram of a Packet Processing System

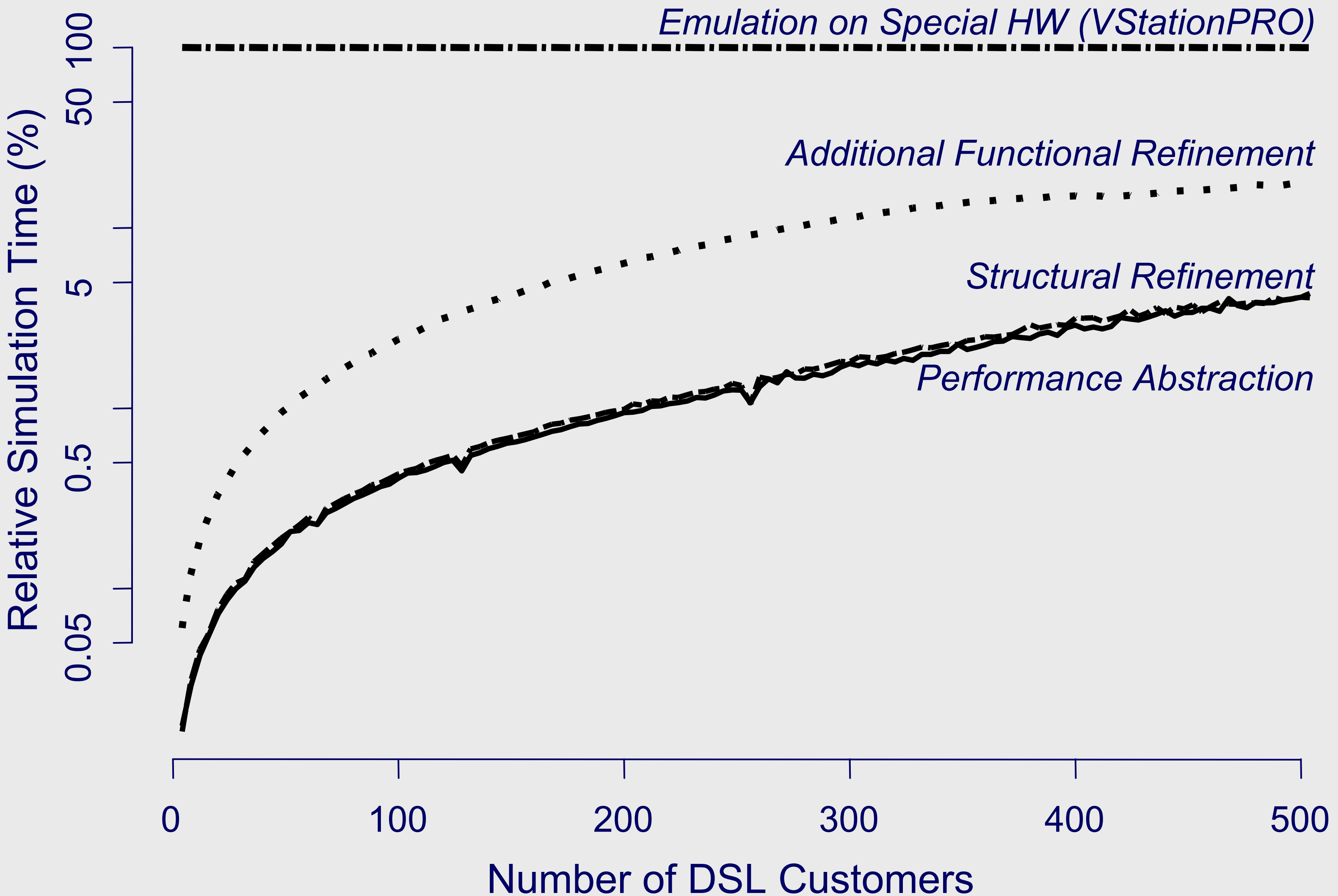


Abstraction Levels and Refinement



Never stop thinking

Simulation Performance of SystemQ



Never stop thinking

Modeling Effort and Quality of Results

- SystemQ simulation environment
  - ◆ 2.4 GHz Intel-based Linux PC
  - ◆ Comparison with Mentor Graphics VStationPRO emulator
- Modeling effort
  - ◆ RTL model to be built within roughly 100 man-weeks
  - ◆ SystemQ model within *one* man-week from scratch
- Quality of results
  - ◆ Quality increases with decreasing abstraction
  - ◆ High abstraction leads to high performance
  - ➔ Balance of performance vs. quality
  - ➔ SystemQ supports different abstraction levels

Conclusion

- SystemQ combines queuing systems and SystemC
- SystemQ features
  - Explicit scheduling
  - Simulation even in concept phase
  - Support of different abstraction levels
  - Path to implementation due to systematic refinement
- Case study revealed
  - Fast simulation
  - Low modeling effort

Never stop thinking

Thank you for your attention!