

Towards Managed Evolution of SDR Systems

Albert Noll

Laboratory for Software Technology, ETH Zurich

Introduction

- Software Defined Radio (SDR) systems substitute many hardware components by software.
- SDR provides additional flexibility with different waveforms, frequencies and channels.
- Hard real-time constraints of SDR systems pose a significant challenge towards the computational capabilities of portable, embedded systems such as mobile phones.

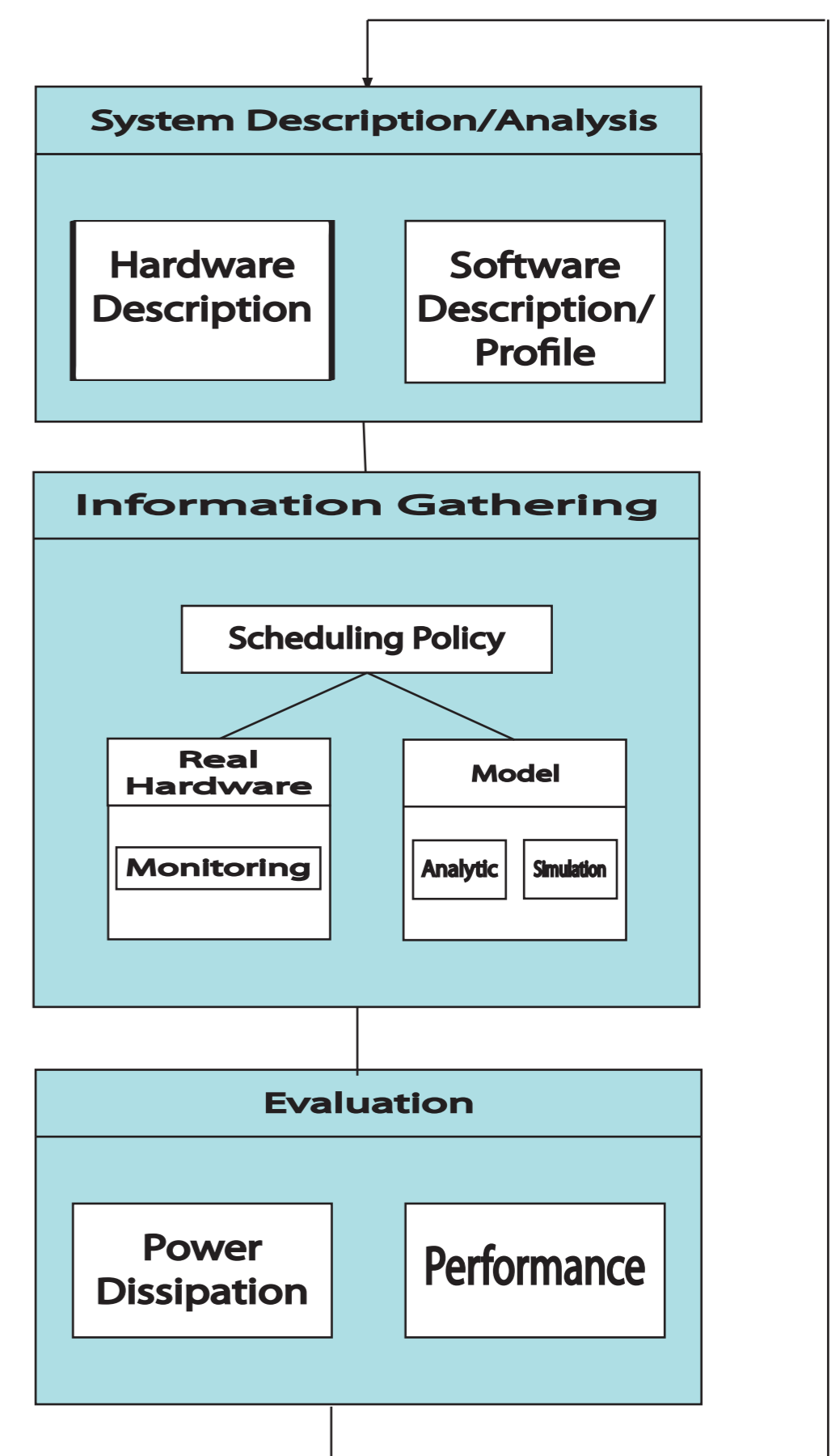
Motivation

- Hardware is becoming more diverse through the introduction of system-on-chip and network-on-chip technologies.
- Short time-to-market periods and low production costs reinforce the re-usability of software components on new hardware architectures.
- A successful migration in product families needs to guarantee that functional and behavioral system specifications are met.

Approach

- Evaluation of system behavior in an early development stage.
- Use existing software and hardware components to estimate behavioral system properties.
- Refine the model with more detailed information as they become available.
- A layered model is well suited to incrementally add more levels of detail to the model.

High-Level Evaluation



The Layered Model

• Layer 0

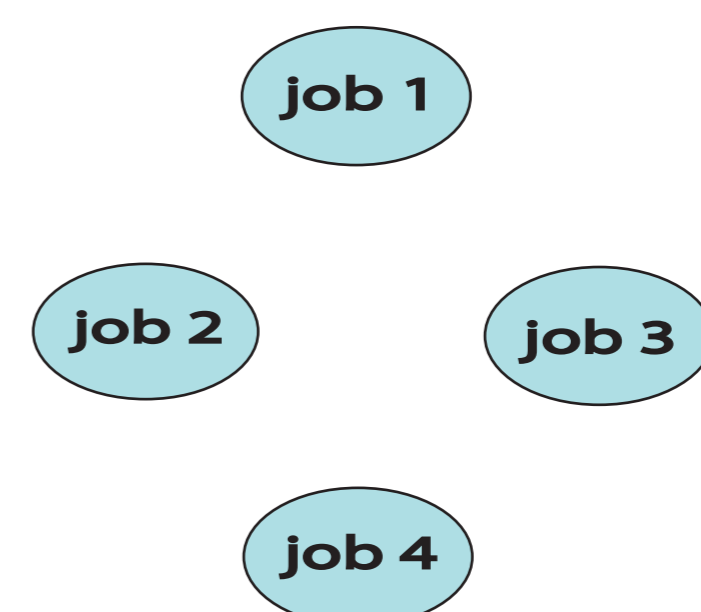
Perspective: set of jobs and various scheduling schemes

Models: static (analytic model) or dynamic (simulation) e.g.: queuing networks

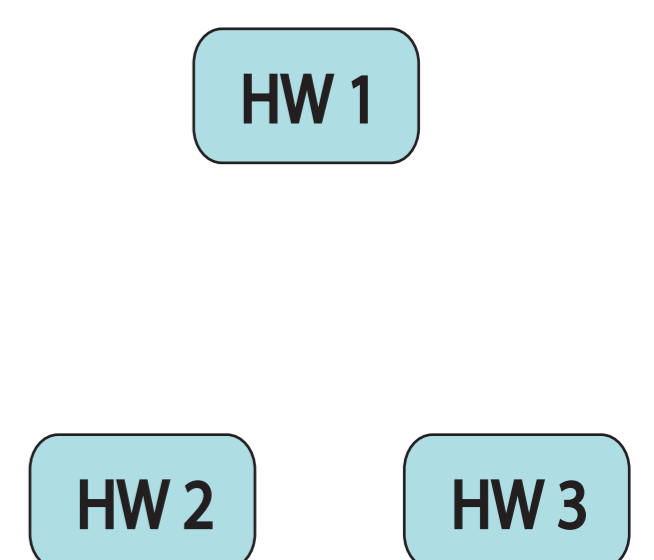
Required Information: - execution times of different jobs on various hardware platforms
 - workload generator (job frequency, number of instances per job, ...)
 - scheduling policy (priority, real-time requirements, ...)

Purpose: evaluation of different hardware platforms for a given set of applications

Software Description



Hardware Description



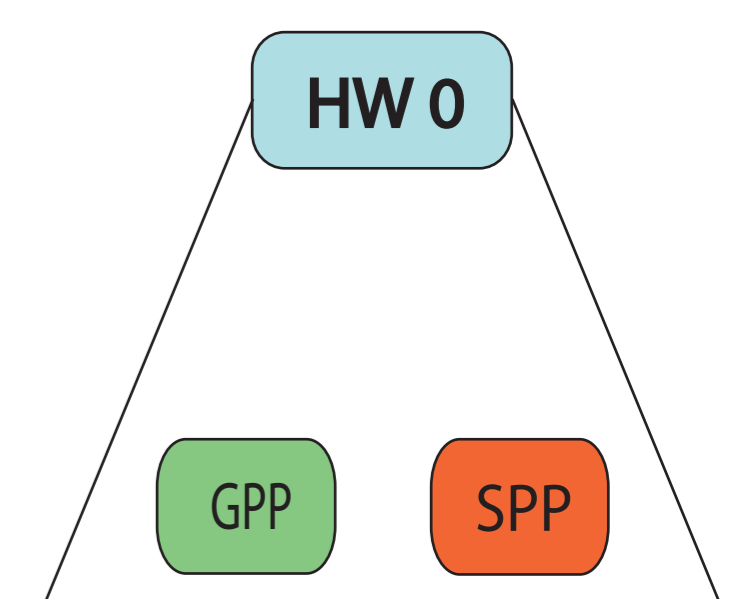
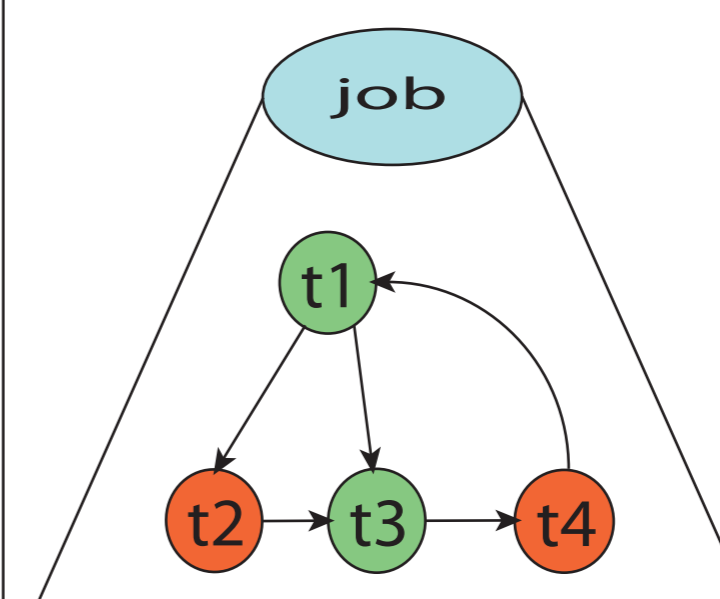
• Layer 1.a

Perspective: data-flow graph and mapping scheme

Model: static or dynamic

Required Information: - execution times of different jobs on various processing units
 - workload generator
 - static or dynamic mapping

Purpose: evaluation of different mapping schemes in a heterogeneous environment



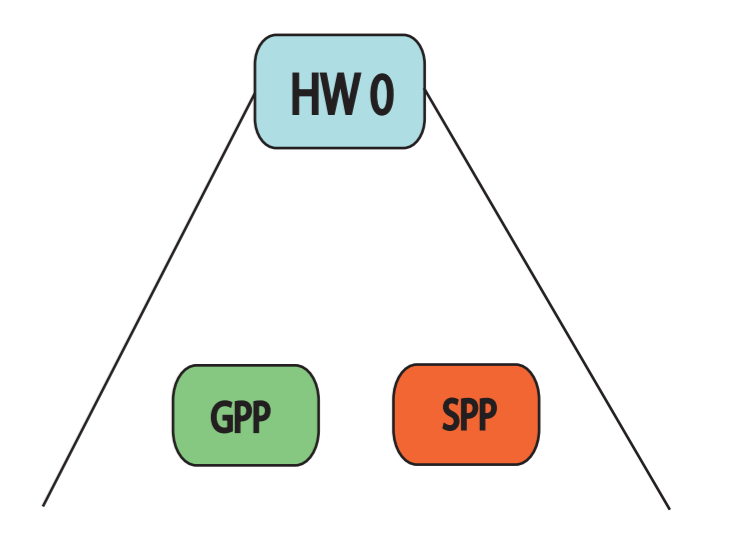
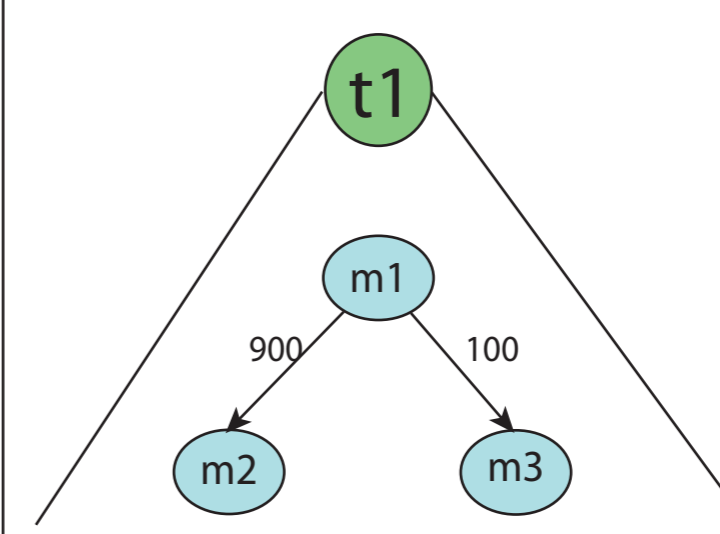
• Layer 1.b

Perspective: method-level

Model: profile-based analysis, automatic generation from software development tools

Required Information: call graph of each task + execution time of each method

Purpose: determine hot-spots for in-depth performance analysis



• Layer 2

Perspective: basic blocks and instructions

Model: trace-based simulation

Required Information: instructions of basic blocks + hardware description of execution unit

Purpose: accurate analysis at a low level + simulation of instruction-level-parallelism

