Taking *Non Functional Properties* into account while studying embedded systems.

Karine Altisen and Matthieu Moy

Verimag

June 11, 2008

Outline

Non Functional Properties

- Physical quantities
  - ex: energy consumption, temperature, time, ...
- Functional Vs Non functional
  - simply observe the value (time, energy, ...) $\rightarrow$ non functional
  - some functionalities depend on this value (QoS) $\rightarrow$ functional
- We focus on simple observation.

Relationships Between Non-Functional Properties

- Time Vs Energy
  - Time is needed to measure energy (integration of power over time)
  - Power-aware systems need the same information for time and energy (energy saving state, DVS ...)
- Energy Vs Temperature
  - The more the chip consumes, the hotest it is, but ...
  - ... the hotest the chip is, the more it consumes.

Outline

Panorama

model	*method*
Detailled description *Include the whole functionality*	Full-simulation *Formal methods not applicable* *Results on some traces*
State-based models *Level of abstraction?* *Faithfulness?*	Model-checking, Abstract interpretation... *State explosion problem* *Exhaustive solution*

Stateless models RTC Scheduling analysis *Faithfulness of the model?*	Analytic solution for *all* executions *Exact solution* *(worst case, best case)*

Panorama

- Intermediate models, mixed analytic and state-based models
  - RTC+Timed automata
  - Probabilistic models, e.g.
    - PEPA — Performance Evaluation for Process Algebra, University of Edinburgh
    - Stochastic Automata Network — performance evaluation for parallel system, Grenoble
  - UPPAAL, Uppsala & Aalborg universities
  - etc

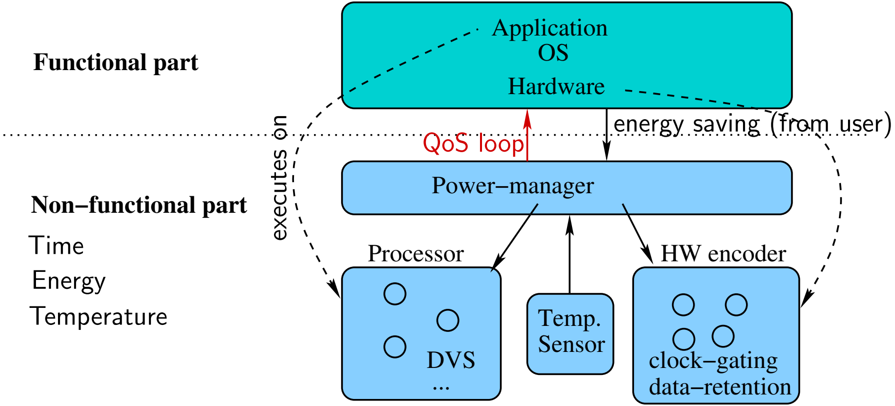
Outline

Sensor Networks and Energy Consumption

- Detailed simulation of the whole network with energy estimation: GLONEMO.
- Based on power-state modeling:
  - Simulates the complete behavior
  - The behavior drives a state machine giving the energy consumption.
- See Florence's talk...

Energy Consumption in SOC

A schematic power-aware system on a chip:



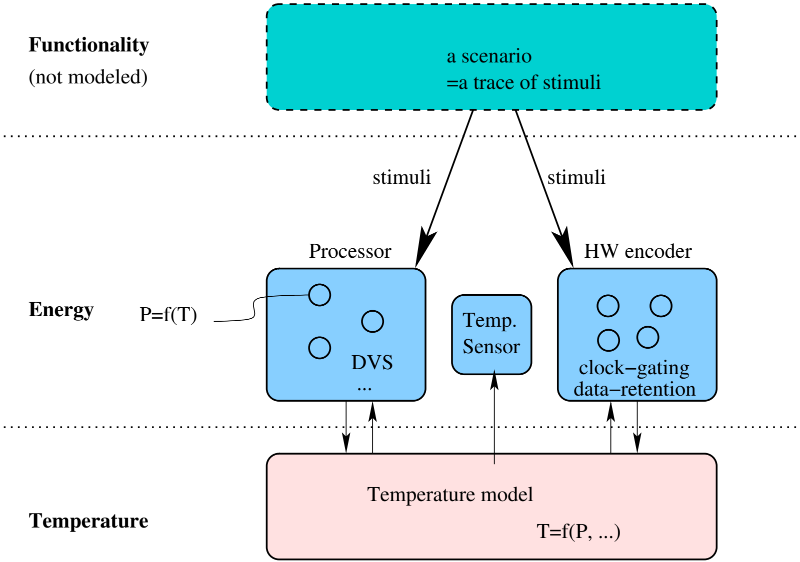
Partnership with Docea-Power

- Local start-up (one end-of-study project in common with them),
- Estimation of energy consumption
- Take ENERGY $\leftrightarrow$ TEMPERATURE into account.

Docea-Power's Tool for Energy and Temperature Estimation

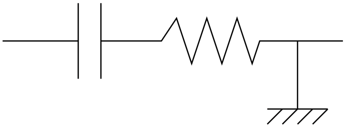
- Model =
  - a state-based model of power consumption mastered by the power manager
  - a temperature model based on the layout of the chip and differential equations
- Estimation of energy consumption =
  - based on "scenarios" (obtained from functional simulation)
  - for one scenario, give the evolution of temperature and energy consumption

Docea-Power Model



Temperature Model

- Based on actual layout on the chip,
- Models heat transfer between pieces of the chip and heat dissipation
- Uses differential equations



- ... we can't help much!

Functionality

- Functional simulation can be done independently, and provides scenarios for the tool,
- Can be RTL simulation or more abstract (TLM),

Power model

- Each component has “power states”
- Notion of transition not described at this level, given by:
  - some inputs for the model that are directly power directive
  - decisions taken by the power-manager
- Power-manager does not appear explicitly:
  - in the actual system, can be HW and/or SW,
  - it implicitly drives HW about power decision
  - by saying which combination of the components power states are possible
  - $\rightarrow$ hand-made product of automata with constraints

Outline

State-Based Behavior is Needed

- Real-life systems have states :
  - Power-manager, systems with energy-saving states
  - Components have distinct power consumptions depending on activity (processor, radio of sensor node, ...)
- Abstracting states
  - would lead to very pessimistic worst case
  - would mean not considering power-saving states!

Solution that Scales Up is Required

Simulation:

- GLONEMO can simulate "in real time" for 1000 nodes
- Docea simulates reasonably slower than real-time (1/15)

Model based approach: (model-checking, abstract interpretation...)

- State explosion problem

Analytic solutions: seems to scale up more easily!

Component-based Solution is Better

- Systems are usually described with components
- e.g. decomposition into functional parts (HW, SW)
- can be an answer to scale up:
  - in terms of description (code reusability)
  - in terms of reasoning (computing the solution)

Outline

Work In Progress

- Work with Yanhong Liu
- A step in your direction?

Work with Yanhong Liu

Models the energy in a sensor node

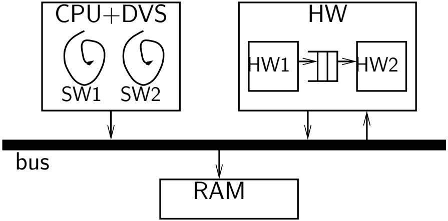
- **Using probabilistic models**
- Attempts to model the MAC radio protocol using PEPA
  Performance Evaluation for Process Algebra, University of Edinburgh
- Experiments of probabilistic model-checking with PRISM (University of Oxford)

Work with Yanhong Liu

- **Using UPPAAL models**
- Models as timed automata:
  - processing elements
  - linked by FIFO queues (with explicit states)
  - + a power manager
- As it is, doesn't scale up
- changing data-granularity may help, but then connecting components is harder.

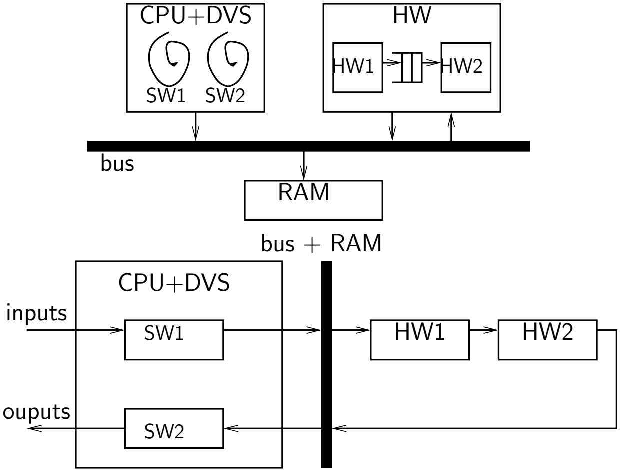
A step in your direction?

A Small Multimedia Example



A step in your direction?

A Small Multimedia Example



A Small Multimedia Example

- **Scenario description:**
  inputs (images to decode) — Proc with DVS: SW1
  — bus+RAM — HW1 –pipeline– HW1 — bus+RAM
  — Proc: SW2 — outputs

- **+ Power Manager:**
  - for each element (SW, HW): several levels of power consumption
    (implies different execution times)

A Small Multimedia Example

- **Scenario description:**
  inputs (images to decode) — Proc with DVS: SW1 — bus+RAM — HW1 –pipeline– HW1 — bus+RAM — Proc: SW2 — outputs
- **+ Power Manager:**
  - for each element (SW, HW): several levels of power consumption
    (implies different execution times)
- *Can MPA-RTC help?* meeting our requirements:

A Small Multimedia Example

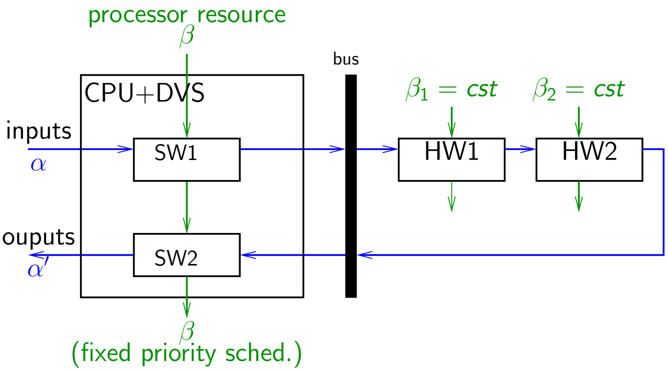
- **Scenario description:**
  inputs (images to decode) — Proc with DVS: SW1 — bus+RAM — HW1 –pipeline– HW1 — bus+RAM — Proc: SW2 — outputs
- **+ Power Manager:**
  - for each element (SW, HW): several levels of power consumption
    (implies different execution times)
- *Can MPA-RTC help?* meeting our requirements:
  1. it scales up,
  2. it seems to fit our notion of component-based model,

A Small Multimedia Example

- **Scenario description:**
  inputs (images to decode) — Proc with DVS: SW1 — bus+RAM — HW1 –pipeline– HW1 — bus+RAM — Proc: SW2 — outputs
- **+ Power Manager:**
  - for each element (SW, HW): several levels of power consumption
    (implies different execution times)
- *Can MPA-RTC help?* meeting our requirements:
  1. it scales up,
  2. it seems to fit our notion of component-based model,
  3. what is the problem exactly? How to model energy?
  4. state-based model?

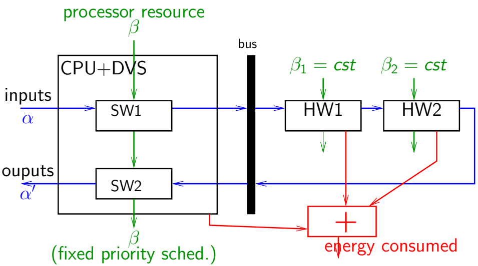
A component-based model

An attempt with MPA-RTC, *but no energy*, *no states*



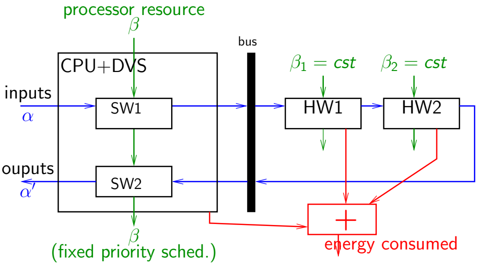
Modeling energy?

Modeling energy?



Problem (e.g.) = what are the *upper and lower bound on power consumption* within a given time interval Δ?

Modeling energy?

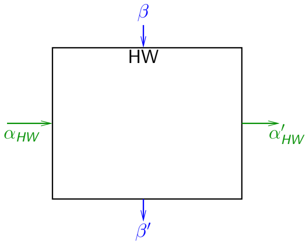


Problem (e.g.) = what are the *upper and lower bound on power consumption* within a given time interval Δ?

Problem (e.g.) = what about the average?

What about states?

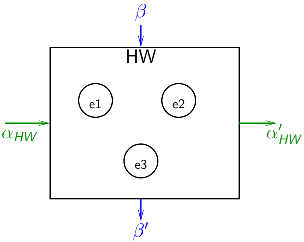
The HW component case:



- available resource depends on state
- moving from a state to another depends on the PM: it is an input of the component

What about states?

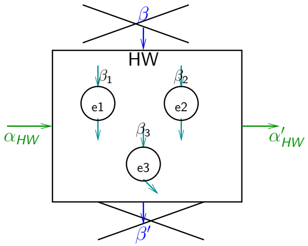
The HW component case:



- available resource depends on state
- moving from a state to another depends on the PM: it is an input of the component

What about states?

The HW component case:



- available resource depends on state
- moving from a state to another depends on the PM: it is an input of the component

Can Event-based Workload Variability help?

This is an open question!

Can Event-based Workload Variability help?

This is an open question!

Event-based workload variability =

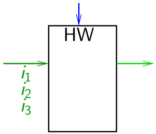
- typed events
  instead of having an arrival curve for one kind of event
- *the workload variability automaton*
  a model for the resource demand of each typed of events
- *the event sequence automata*
  a model of the environment that sends the inputs

Can Event-based Workload Variability help?

This is an open question!

Can Event-based Workload Variability help?

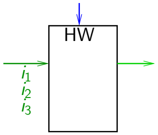
This is an open question!

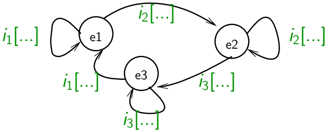


Inputs are typed, depending on power states:

- one type per power state
- $\longrightarrow$ their resource demand is computed depending on the state

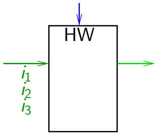
Can Event-based Workload Variability help?

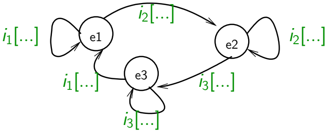




Inputs are typed, depending on power states.

Can Event-based Workload Variability help?



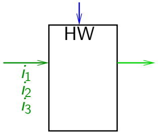


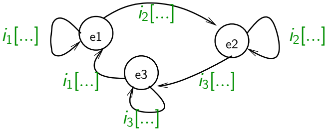
Inputs are typed, depending on power states.

Need for a model of the resource demand of the inputs:
(the workload variability automaton)

- $\longrightarrow$ the states of the HW component
- + transition from the power manager:
  express the resource demand of each input,
  depending on their type (ie their state consumption)

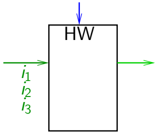
Can Event-based Workload Variability help?

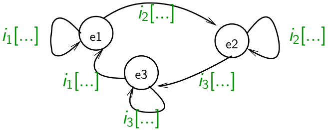




Inputs are typed, depending on power states.
Need for a model of the arrivals.

Can Event-based Workload Variability help?





Inputs are typed, depending on power states.
Need for a model of the arrivals.

Need for a model of the environment:
(the event sequence automata)

- how the inputs are sent? (express by a regexp)