Using UPPAAL for performance modeling & analysis

Martijn Hendriks & Marcel Verhoef
Radboud University Nijmegen
first.last@cs.ru.nl

Agenda

- Context
- Modeling computation
- Modeling communication
- Modeling the environment
- Composing the system
- Analysis of the model
- Some results & conclusions

Context

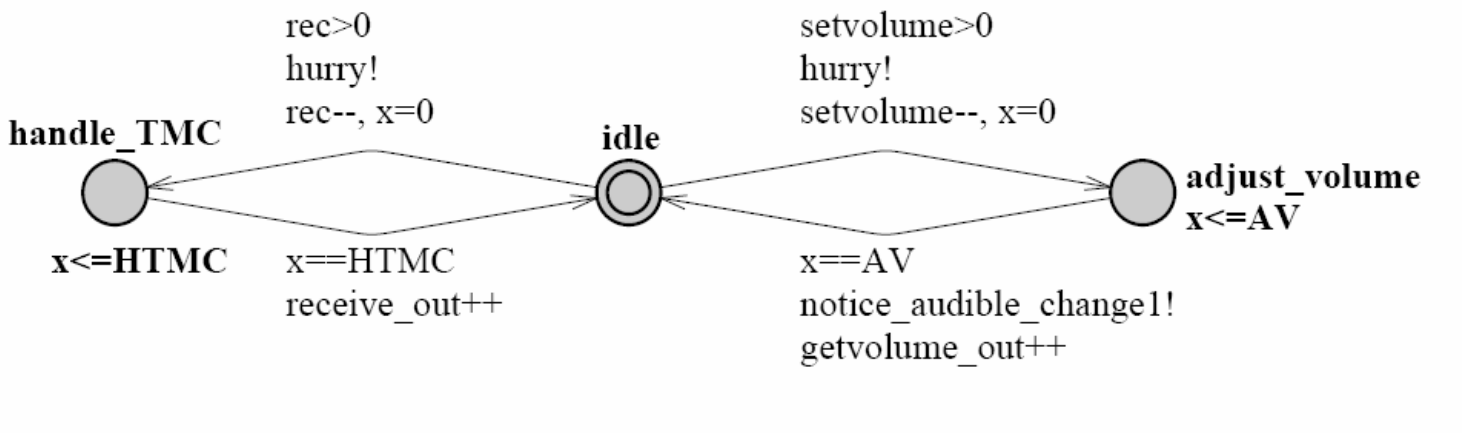
- Early design exploration
- Abstract models
- In-car Radio Navigation system case
- Predict best- and worst case execution times

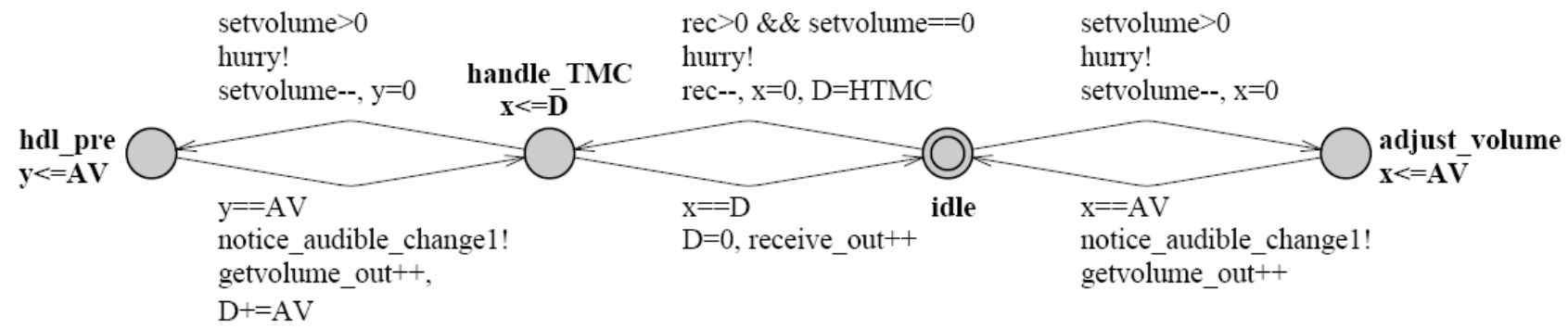
- Questions
  - Can we model the case effectively?
  - Can we analyze the model efficiently?
  - How useful are the results?

Modeling computation resources

- TA per computation resource
- Build list of all operations that the resource performs
- TA is specific for a given deployment
- Resource is either idle or performing some operation
- Resource state is modeled as a location in the TA
- Time spend in location is #instr / capacity
- “greedy” automaton to ensure finite response times
- Count number of outstanding requests per operation
- Scheduling can be modeled (i.e. preemption)

Modeling computation resources





Modeling communication resources

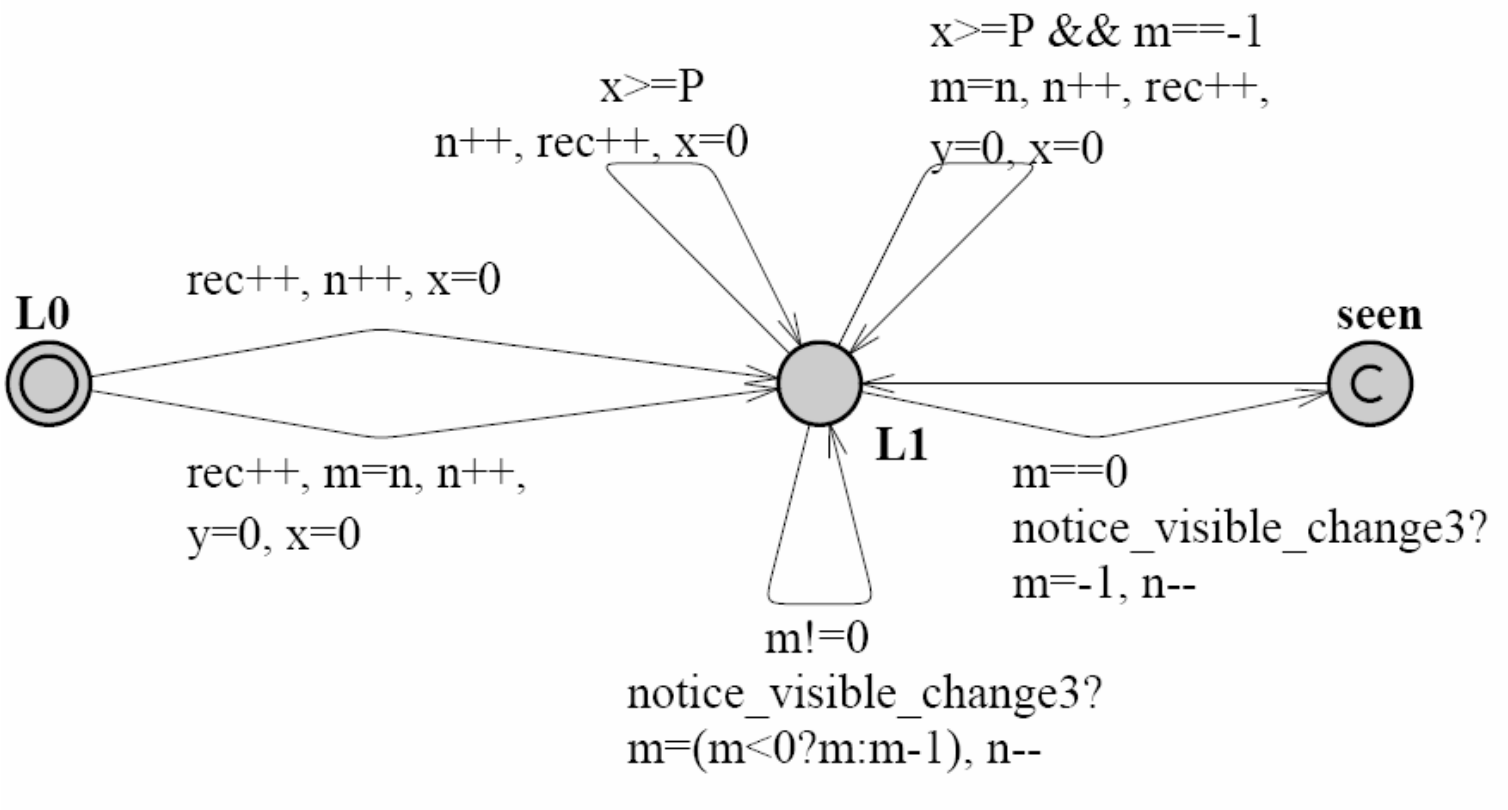
- TA per communication resources
- Build list of all messages that might be transported
- TA is specific for a given deployment
- Resource is either idle or transfering a message
- Resource state is modeled as a location in the TA
- Time spend in location is #size / bandwidth
- “greedy” automaton to ensure finite response times
- Count number of outstanding transfer requests per message
- Bus behavior can be modeled (e.g. priorities)

Modeling the environment

- Template TAs; supported event models:
  - Periodic
  - Periodic with offset (phase shift)
  - Sporadic
  - Periodic with jitter (j < p)
  - Bursty (j >> p) with minimum inter arrival time

- Two flavours
  - event generators
  - event generator with measuring capability (assumption: order preserving - fifo behavior)

Modeling the environment



Modeling the system

- Simply compose the system model by
  - TAs for all computation resources +
  - TAs for all communication resources +
  - Event generator TAs +
  - Measuring event generator TA +
  - “hurry” automaton

Performing the analysis

- AG (aut.seen → aut.y < C)
- Perform binary search (manually)
- Results typically found in a few seconds or
- Use search strategy: find any bound
- “property not satisfied” → counter example
- Only [BW]CET analysis, no utilisation

Observations (1)

- Found some useful modular modeling strategies
- Model construction is currently manual process
  laborious and error prone
- We believe that automation (model construction)
  is possible
- Analysis of this size of case study is possible
  results are found within seconds, minutes rather than hours
- Results found comparable (competitive) to other techniques

Observations (2)

- State space explosion problem is still present, mainly determined by
  - size of the model
  - difference in clock periods of environment model (ms, sec)
  - level of non-determinism in the model

- Can be (partly) circumvented by
  - Smart modeling (expert use of uppaal)
  - Use UPPAAL for non-exhaustive search (using search strategies); find *any* value (lower bound)