Timed Automata Based Analysis of Embedded System Architectures

Martijn Hendriks & Marcel Verhoef
Radboud University Nijmegen (NL)
***first.last*@cs.ru.nl**

Agenda

- Background & rationale to the work
- The in-car radio navigation case study
- Constructing timed automata models
- Comparison to other techniques
- Conclusions

Context

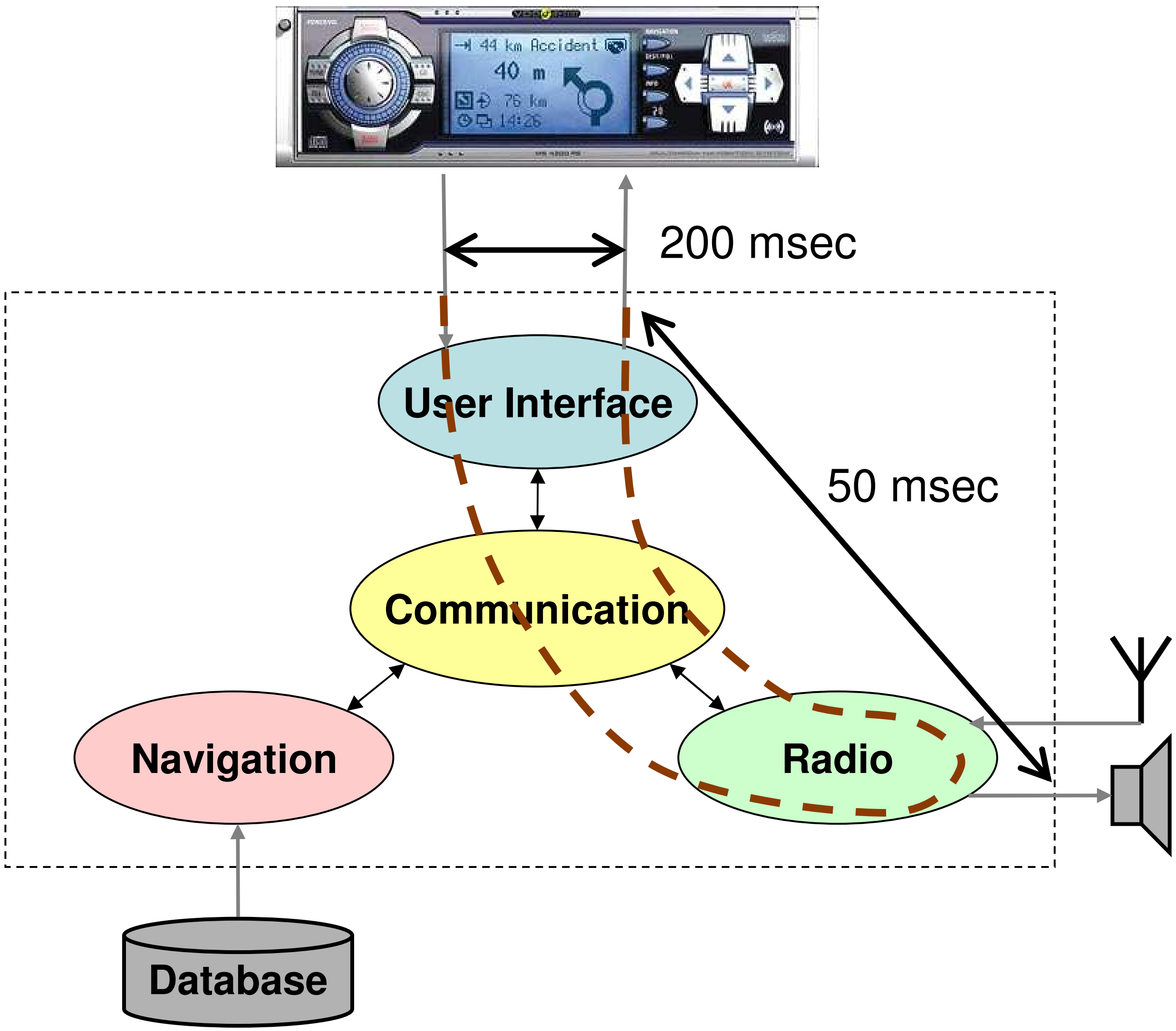
- Early design exploration, abstract models
- UPPAAL model checker increasingly powerful
- In-car Radio Navigation system case study
- Predict best- and worst case execution times

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

The In-Car Radio Navigation System

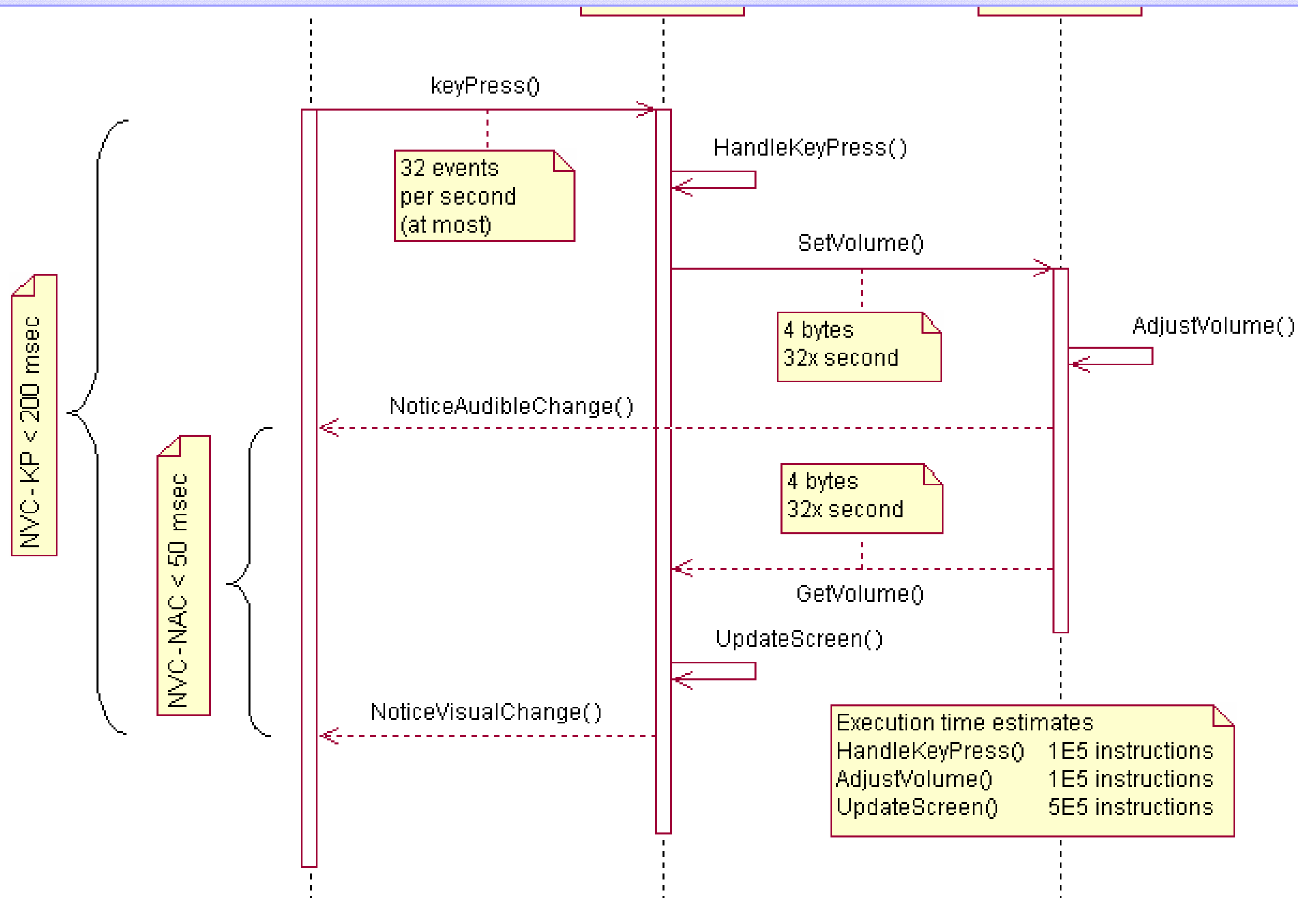
- Car radio with built-in navigation system
- User interface needs to be responsive
- Traffic messages must be processed in a timely way
- Several applications may execute concurrently

System Overview – Change Volume

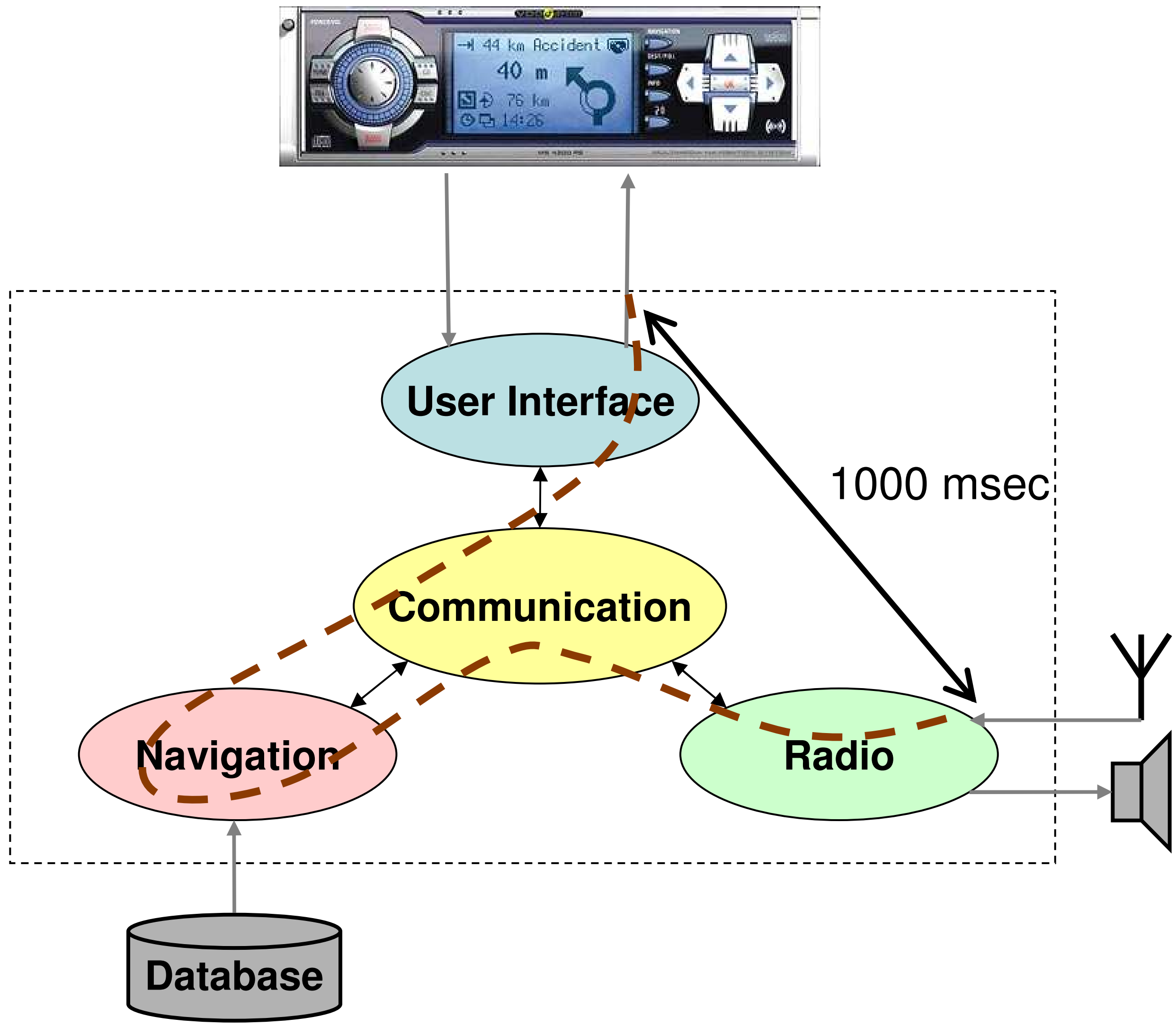


Application A: Change Audio Volume

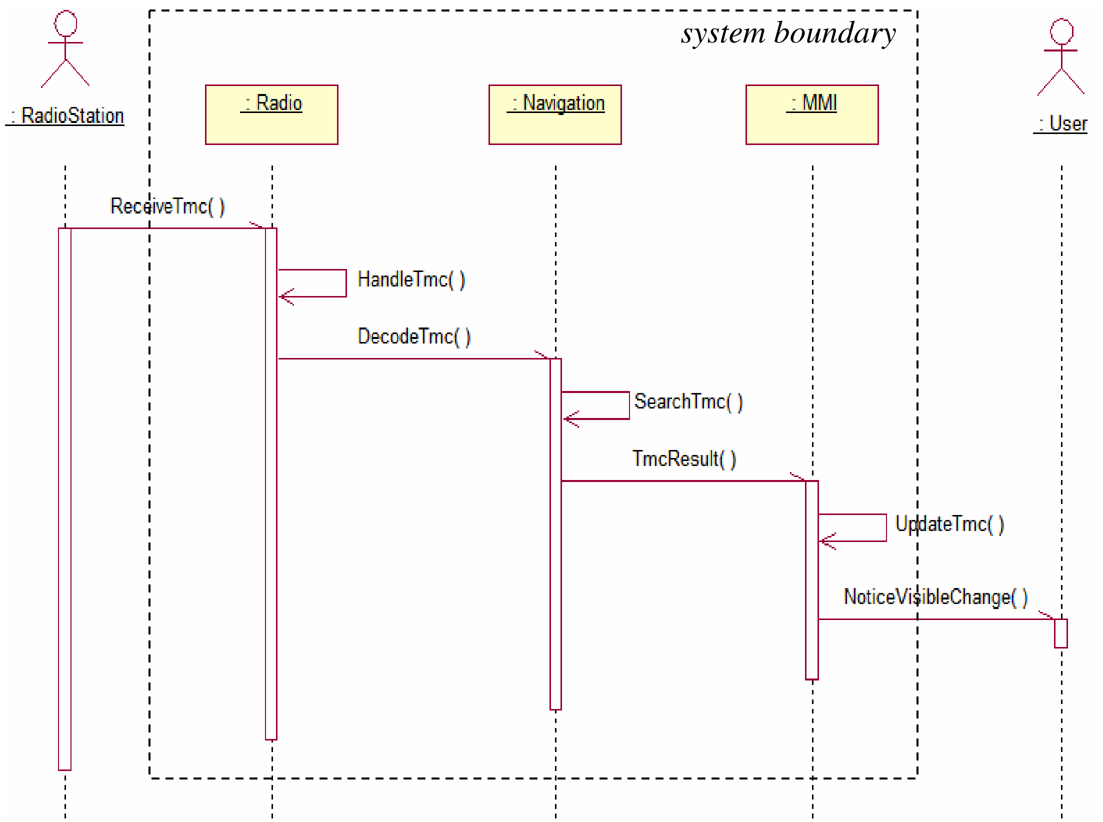
Computation Resource Demand



System Overview – Handle TMC



Application C : Handle TMC



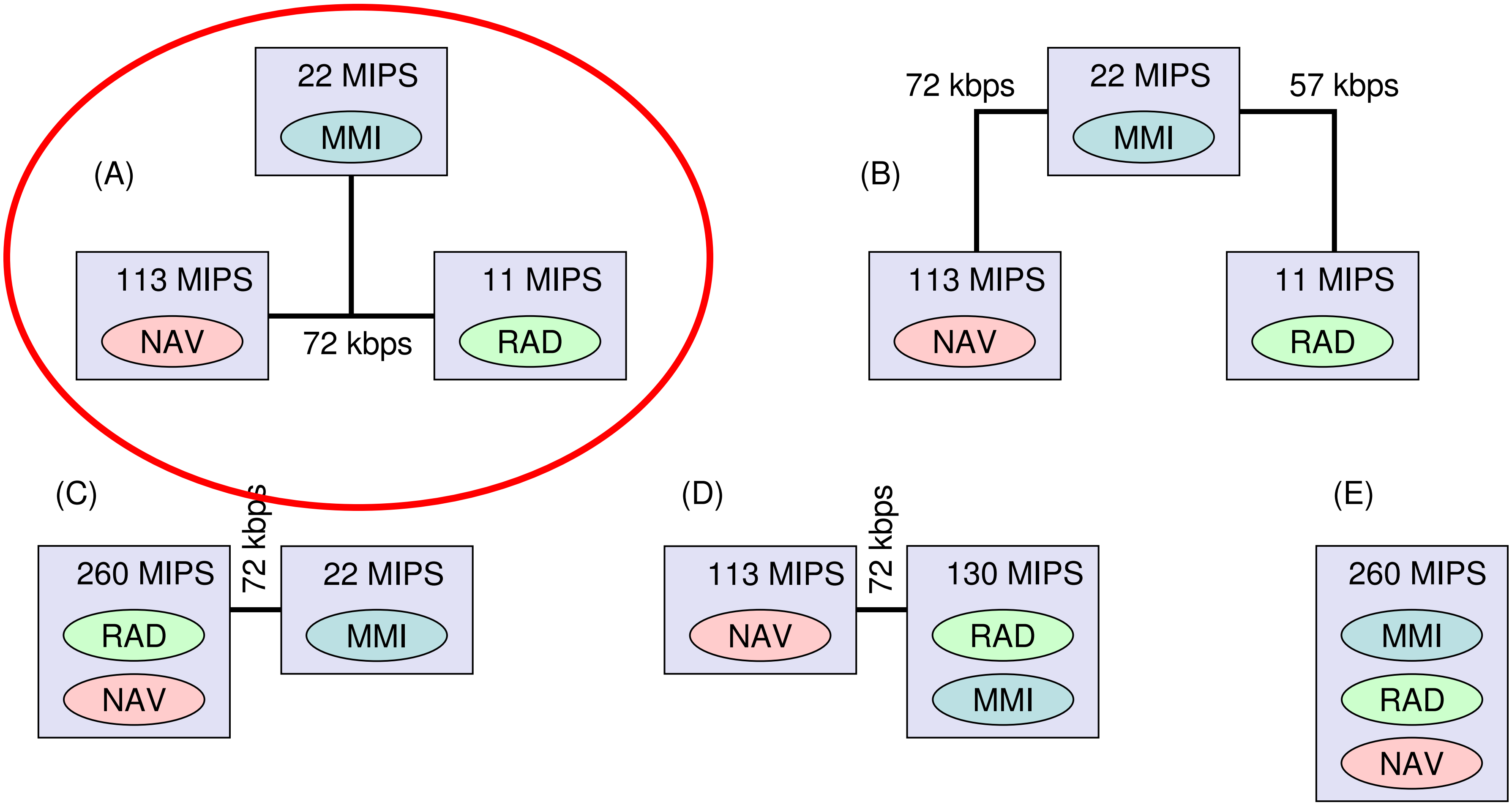
ReceiveTmc:
pure periodic 0.333 Hz, jitter 0

TASKS (priority[1], #instructions)
HandleTmc, 4, 1E6
SearchTmc, 5, 5E6
UpdateTmc, 6, 5E5

MESSAGES (priority[1], #size)
DecodeTmc, 4, 64 bytes
TmcResult, 5, 64 bytes

REQUIREMENTS
NVC – ReceiveTMC ≤ 1000 msec

Proposed Architecture Alternatives



- kbps = kilo bit per second
- mips = 10^6 instructions per second
- assume no (protocol or scheduling) overhead (zero cost)
- inter task communication on same resource is instantaneous (zero cost)

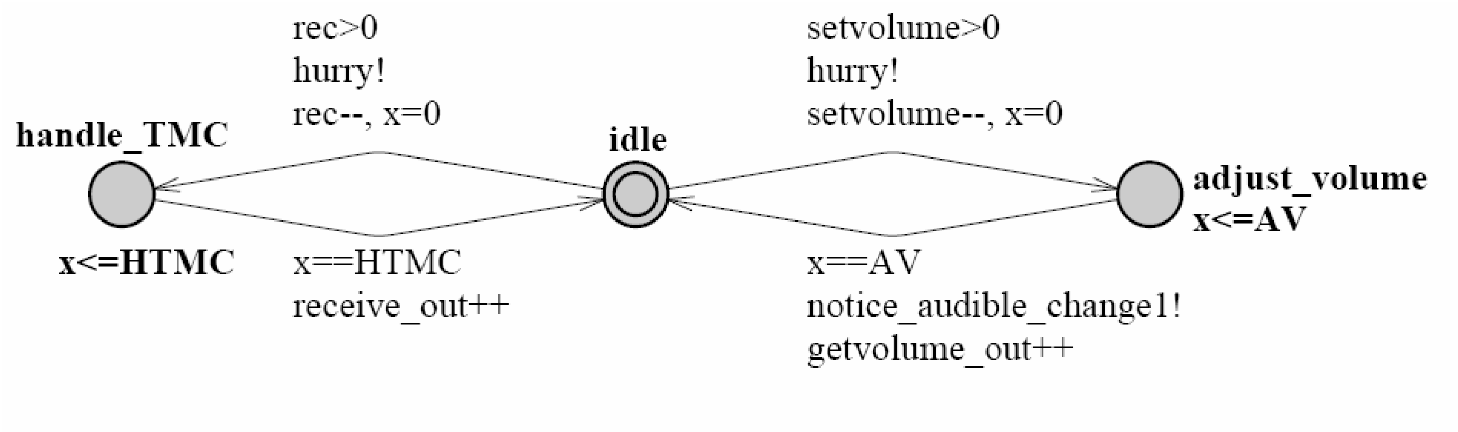
Constructing Timed Automata

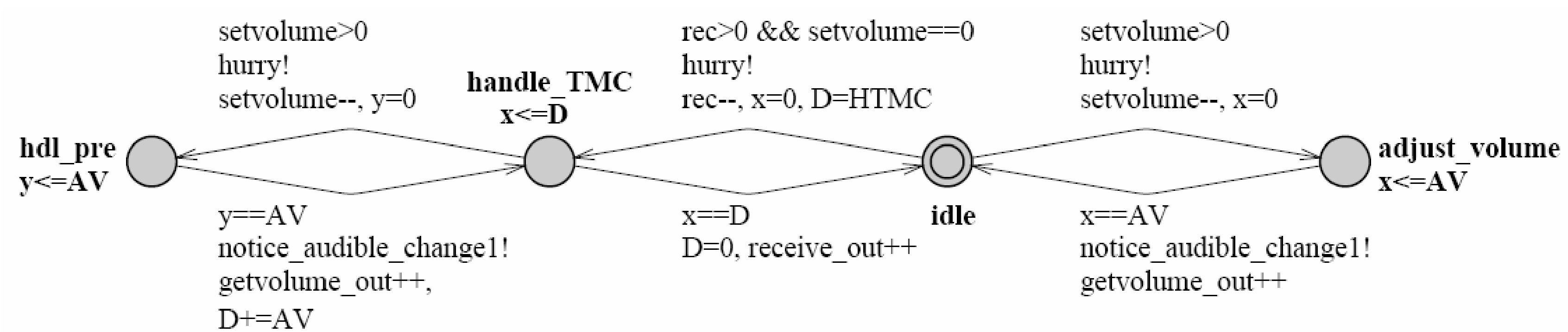
- Modeling computation resources
- Modeling communication resources
- Modeling the environment
- Composing the model

Modeling computation (1)

- 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 (2)





Modeling communication

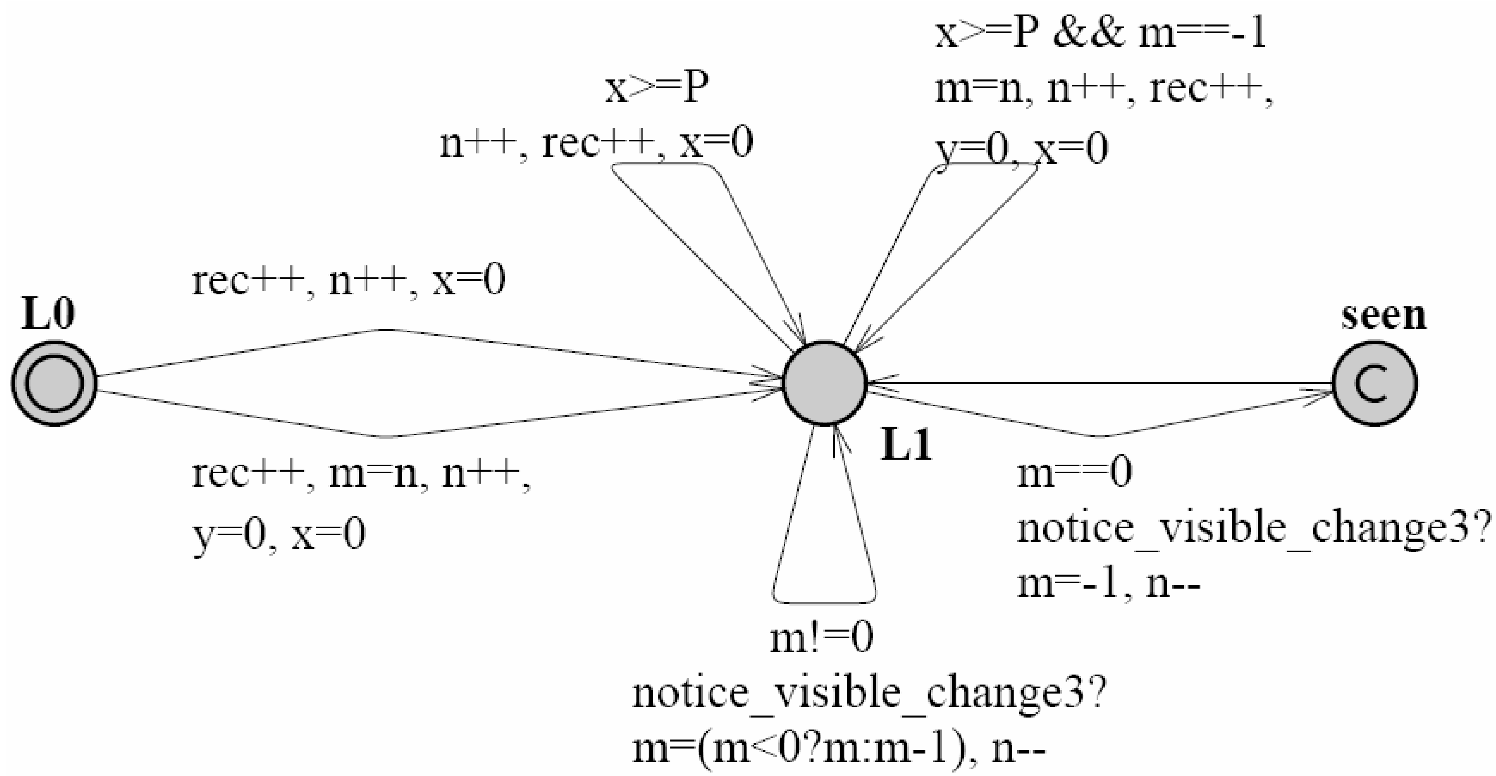
- TA per communication resources
- Build list of all messages that might be transported
- TA is specific for a given deployment
- Resource is either <u>idle</u> or <u>transferring a message</u>
- Resource state is modeled as a location in the TA
- Time spend in location is #size / bandwidth
- “greedy”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

Modeling the environment (1)

- 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 flavors
  - event generators
  - event generator with measuring capability (assumption: order preserving - FIFO behavior)

Modeling the environment (2)



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 (1)

- 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 utilization

Performing the analysis (2)

Table 1. Uppaal worst-case response time analysis results (in milliseconds)

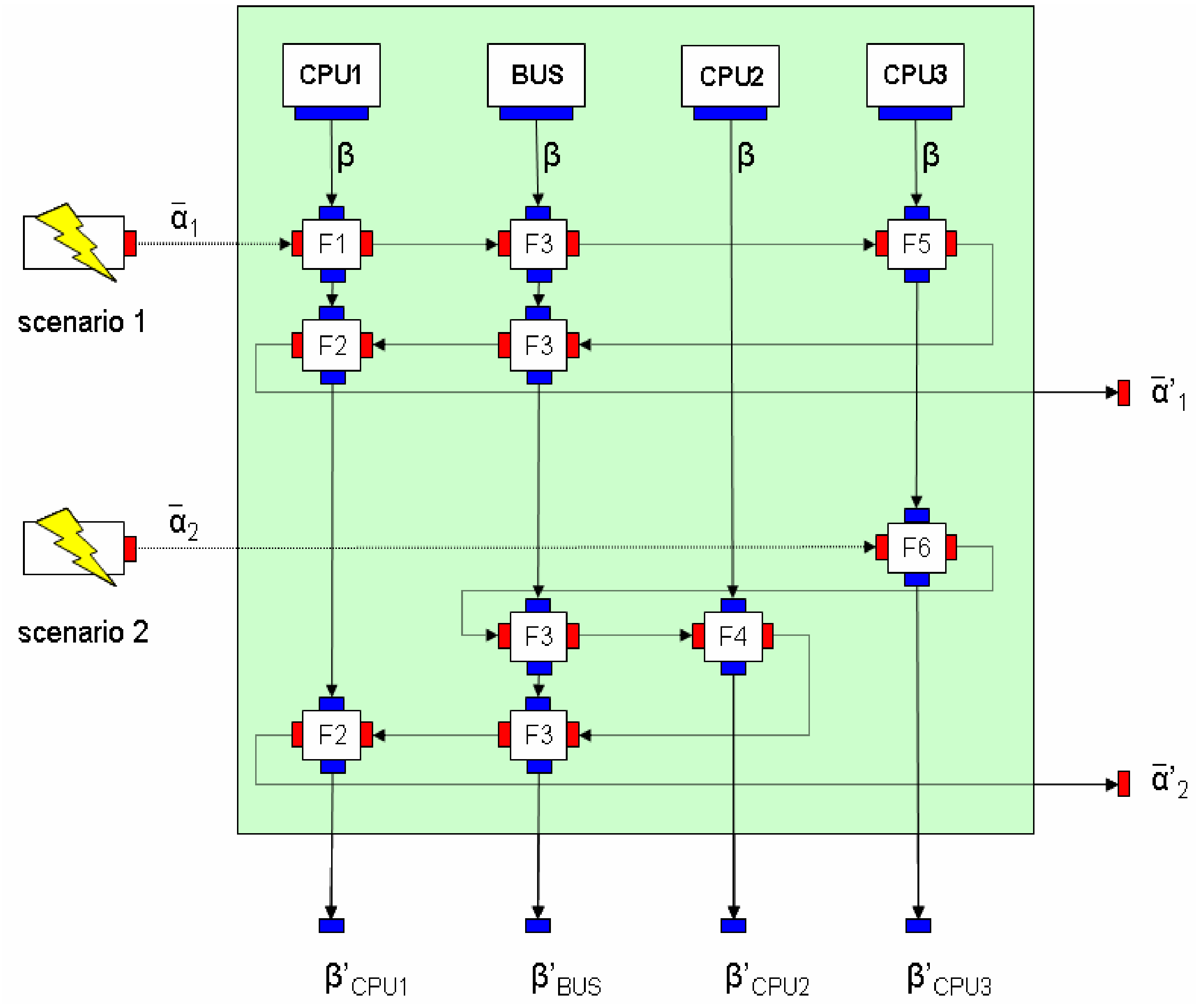
Requirement \ *Event model*	$po\ (F=0)$	pno	sp	$pj\ (J=P)$	$bur\ (J=2P, D=0)$
HandleTMC (+ ChangeVolume)	357.133	381.632	382.076	$>$ 400.000 (df)	$>$ 500.000 (rdf)
HandleTMC (+ AddressLookup)	172.106	239.080	239.080	329.989	420.898
K2A (ChangeVolume + HandleTMC)	27.716	27.716	27.716	$>$ 27.715 (bf)	$>$ 27.715 (bf)
A2V (ChangeVolume + HandleTMC)	41.796	41.796	41.796	$>$ 41.795 (bf)	$>$ 41.795 (bf)
AddressLookup (+ HandleTMC)	79.075	79.075	79.075	79.075	79.075

Comparison: MPA (1)

- Modular Performance Analysis
- Developed at ETH Zurich (Lothar Thiele et al)
- Performance networks analysed with real-time calculus
- Analytic method, deterministic queuing theory
- Adaption of Network Calculus (Boudec, Thiran)
- Describes event streams by interval bound functions
- Information is lost: $t \rightarrow \Delta t$
- Evaluation is very fast (no simulation)

- http://www.mpa.ethz.ch

Comparison: MPA (2)

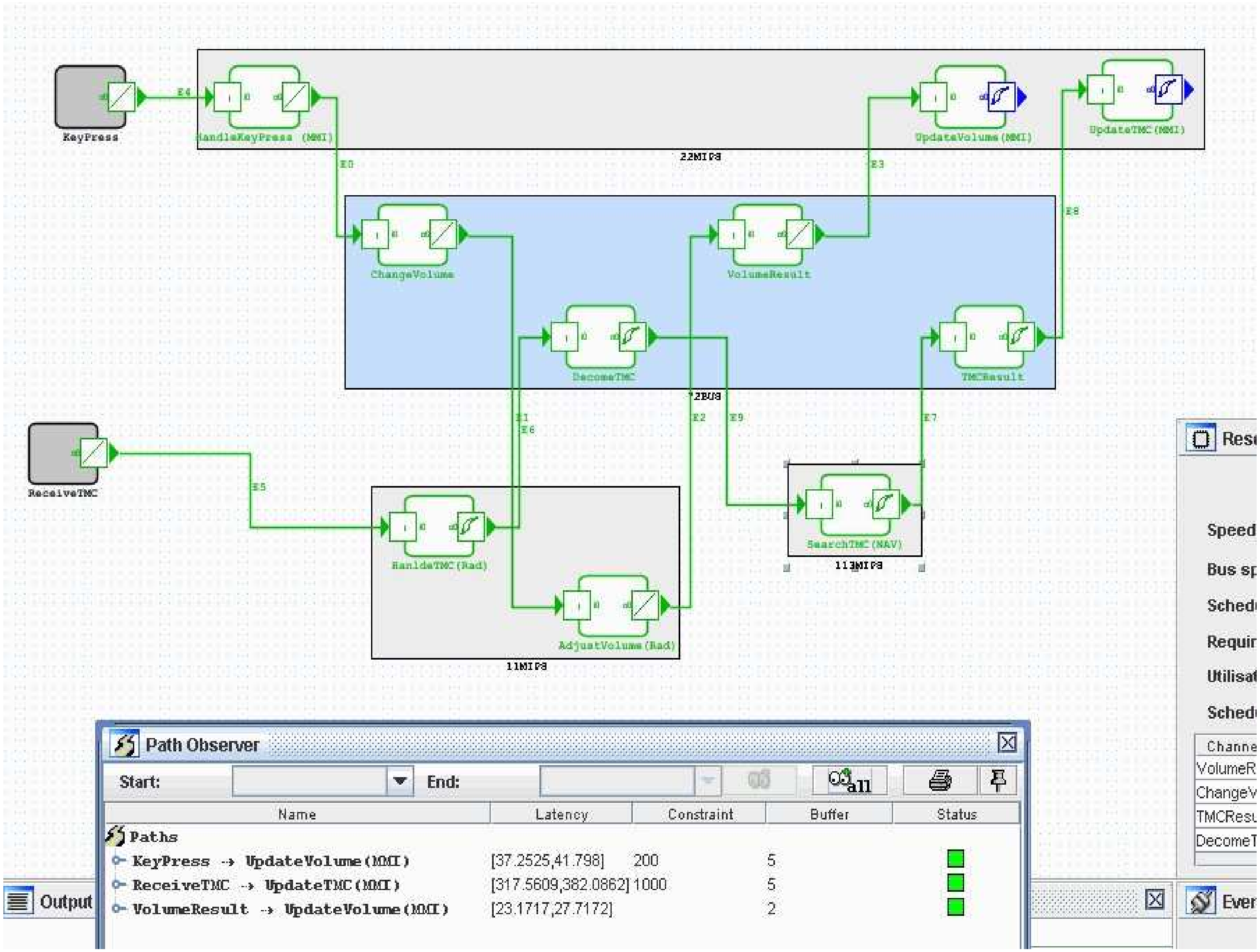


Comparison: SymTA/S (1)

- Symbolic Timing Analysis for Systems
- Developed at TU Braunschweich (Rolf Ernst et al)
- Classical (formal) scheduling analysis techniques
- Symbolic simulation
- Calculate resource local optima
- Optimize system level by iteration over local optima
- Heterogeneous architectures
- Complex task dependancies, context aware analysis
- Rapid design space exploration by sensitivity analysis

- http://www.symtavision.com

Comparison: SymTA/S (2)



Comparison with MPA and SymTA/S

Table 2. Worst-case response time results – comparison with other tools

Requirement \ *Tool*	*Uppaal (po)*	*Uppaal (pno)*	*SymTA/S (pno)*	*MPA (pno)*
HandleTMC (+ ChangeVolume)	357.133	381.632	382.086	390.0862
HandleTMC (+ AddressLookup)	172.106	239.080	253.304	265.8491
K2A (ChangeVolume + HandleTMC)	27.716	27.716	27.717	28.1616
A2V (ChangeVolume + HandleTMC)	41.796	41.796	41.798	42.2424
AddressLookup (+ HandleTMC)	79.075	79.075	79.076	84.066

- many thanks to Ernesto Wandeler (MPA) and Kai Richter (SymTA/S)

Conclusions (1)

- Found some useful modeling strategies
- Model construction is currently manual process laborious and error prone
- We believe that model construction can be automated
- 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

Conclusions (2)

- State space explosion is still likely, determined by
  - size of the model
  - difference in clock periods of environment model
  - 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)

Thank you for your attention!

- Some additional on-line resources
  - UPPAAL model checker
    http://www.uppaal.com
  - The AMETIST project
    http://ametist.cs.utwente.nl
  - UPPAAL models of this case study
    http://www.cs.ru.nl/~martijnh/
  - General case study description
    http://www.tik.ee.ethz.ch/~leiden05