

TIMES: a Tool for Schedulability Analysis and Code Generation of Real-Time Systems

Tobias Amnell Elena Fersman Leonid Mokrushin
Paul Pettersson Wang Yi

Presenter: Henrik Kragh-Hansen

September 24, 2007

TIMES

Why real-time systems?

- Expensive failures
- Loss of lives
- Environmental damages
- Financial loss

TIMES

Features

- Schedulability analysis
- Verification
- Code generation

TIMES

Schedulability analysis

- Worst case response time
- All tasks are computed within their deadlines
- Reachability problem
- Decidable

Benefits

- Ensures deadlines
- Prevent failure

TIMES

Schedulability analysis

- Worst case response time
- All tasks are computed within their deadlines
- Reachability problem
- Decidable

Benefits

- Ensures deadlines
- Prevent failure

TIMES

Code generation

- Executable C-code
- Tasks does not exceed worst case execution time

Benefits

- Ensures valid code
- Reduces development time
- Limits inconsistency

TIMES

Code generation

- Executable C-code
- Tasks does not exceed worst case execution time

Benefits

- Ensures valid code
- Reduces development time
- Limits inconsistency

Outline

- 1 Introduction
- 2 Terminology
- 3 TIMES
- 4 Conclusion

Terminology

Two central concepts

- Task - an executable program
- Task model - a task arrival pattern

Task

Types of tasks

- Periodic
- Sporadic
- Preemptive or Non-preemptive

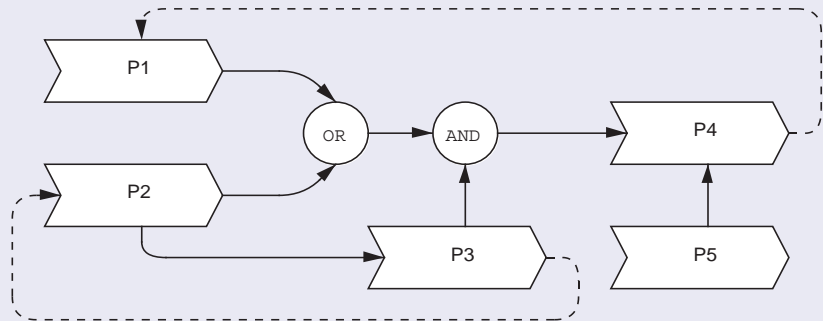
Tasks Parameters and Constraints

Timing constraints

- Relative deadline (D)
- Worst case execution time are pre-specified (C)
- $C \leq D$
- $[C_B, C_W]$

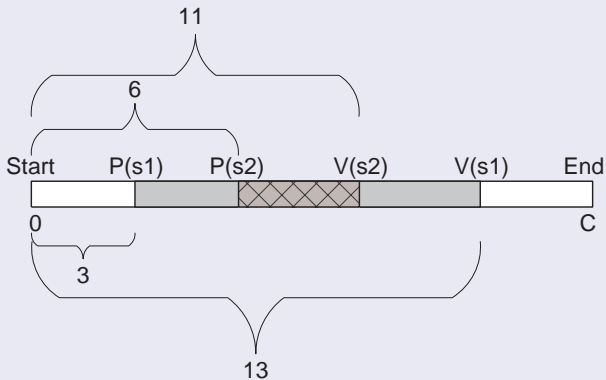
Tasks Parameters and Constraints

Precedence constraints



Tasks Parameters and Constraints

Resource constraints



Timed Automata as Task Arrival Patterns

Timed automata extended with variables and tasks

- Edge labeled with:
 - Guard
 - Sync
 - Assign/Update
- Locations may be annotated with task(s)
 - Task queue
 - Scheduled according to strategy

TIMES GUI

C:\SporadicPeriodic.xml - TimesTool

File Run Options Window Help

SporadicPeriodic Task Automaton_A

Scheduling policy: Deadline Monotonic Preemptive

Name	Pri	C	D	T
task_A	4	1	3	<input checked="" type="checkbox"/>
task_B	3	5	20	<input type="checkbox"/>
task_C	2	8	28	<input type="checkbox"/>
task_D	1	5	30	<input type="checkbox"/>

All Periodic Non-periodic

Properties

Template attributes

Name: Automaton_A

Parameters: const N

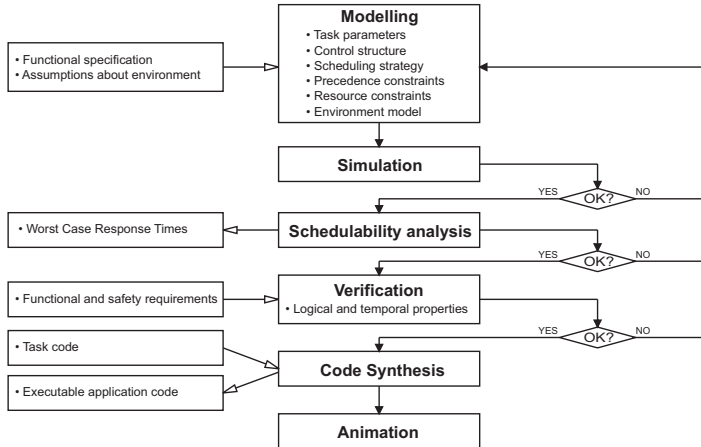
Environment

Local declarations

Name	Type	Value
x	clock	0
y	clock	0

Ready

Design Process using TIMES



Schedulability Analysis

Timed automata extended with tasks

- A semantic state is a triple (l, u, q)
- l current control location
- u current values of clocks and data variables
- q current task queue

Schedulability Analysis

Non-schedulable state

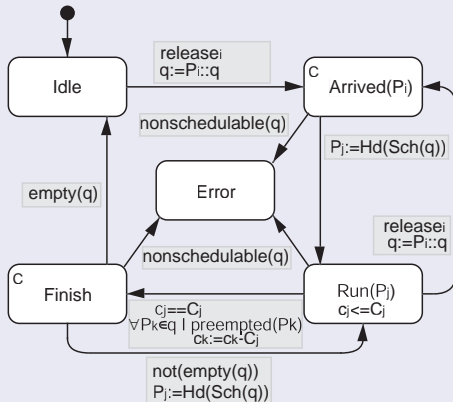
- The state (l, u, q)
- q contains a task with a missed deadline

Schedulable?

- An automaton is either schedulable or non-schedulable
- Reachability problem
- Decidable

Schedulability Analysis

Scheduler automaton



Schedulability Analysis

Schedulability strategy

- Highest priority first (FPS)
- First come first served (FCFS)
- Earliest deadline first (EDF)
- Least laxity first (LLF)

Code Generation

The process

- Automata extended with tasks as design models
- Validated design model \Rightarrow executable code
- Preserves behavior of model.
- Platform must correspond to requirements

Target platform

- Generated for brickOS used for LEGO Mindstorm
- Tasks as separate threads

Is TIMES Actually Used?

A non-trivial application:

- Control software of a production cell
- 12 tasks
- 7 automata
- 17 integers
- 24 booleans
- 31 clock variables (7 in model and 24 in scheduler)

Results

- Two 1.8 GHz AMD, 2GB RAM, running Mandrake Linux
- 207 MB memory and terminates in 11 minutes
- Using over approximation only uses 13 MB and 9 seconds

Is TIMES Actually Used?

UML SPT profile

- Extension to the UML standard
- Model time and time-related aspects of embedded systems

Rhapsody developed by I-Logix (Telelogic)

- TIMES as a plug-in tool for schedulability
- Embedded Market Forecasters Rank Rhapsody #1 for Productivity

Contributions of TIMES

Two main functions

- Schedulability analysis
- Code generation

My Opinion

Pros

- Extends UPPAAL
- Schedulability analysis
- Model \Rightarrow Code

Cons

- Missing functionality UPPAAL
- How to retrieve worst case execution time?
- Code \Rightarrow Model

My Opinion

Pros

- Extends UPPAAL
- Schedulability analysis
- Model \Rightarrow Code

Cons

- Missing functionality UPPAAL
- How to retrieve worst case execution time?
- Code \Rightarrow Model

Thanks for Your Attention

Questions?