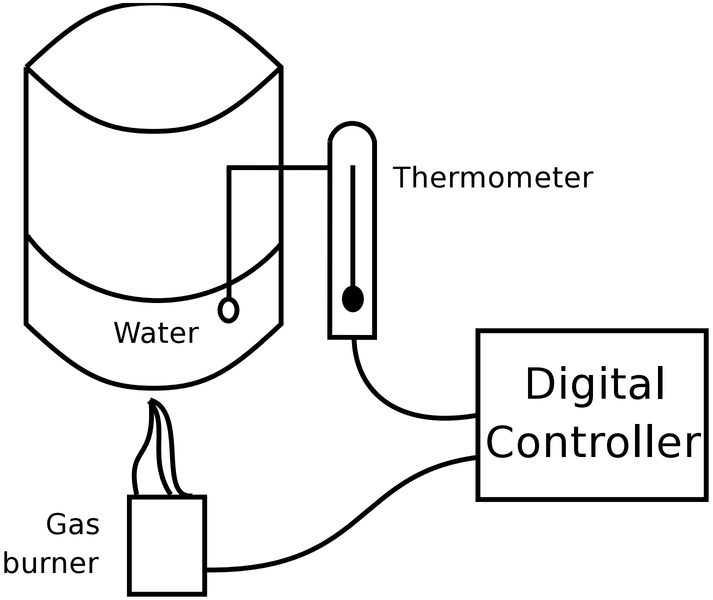
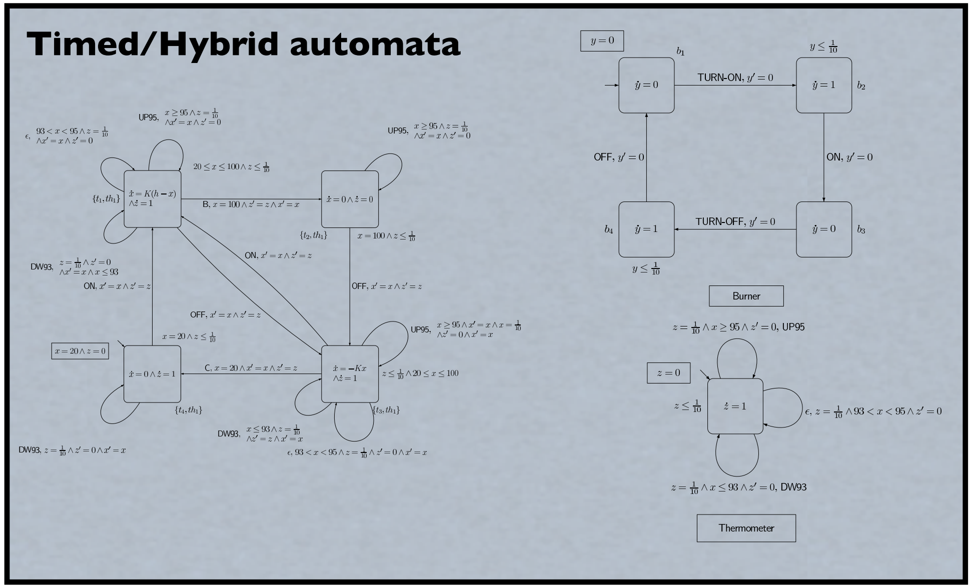
Controllers:
Robustness and Synthesis

Prof. Jean-François Raskin
CFV - Université Libre de Bruxelles

ARTIST 2 - Cluster on
Verification and Testing

Model Based Development




Timed/Hybrid automata

UP95, $x \geq 95 \wedge z = \frac{1}{10} \wedge x' = x \wedge z' = 0$
ϵ, $93 < x < 95 \wedge z = \frac{1}{10} \wedge x' = x \wedge z' = 0$
$20 \leq x \leq 100 \wedge z \leq \frac{1}{10}$
$\{t_1, th_1\}$
$\dot{x} = K(h - x) \wedge \dot{z} = 1$
B, $x = 100 \wedge z' = z \wedge x' = x$
$\dot{x} = 0 \wedge \dot{z} = 0$
UP95, $x \geq 95 \wedge z = \frac{1}{10} \wedge x' = x \wedge z' = 0$
$\{t_2, th_1\}$
$x = 100 \wedge z \leq \frac{1}{10}$
ON, $x' = x \wedge z' = z$
DW93, $z = \frac{1}{10} \wedge z' = 0 \wedge x' = x \wedge x \leq 93$
ON, $x' = x \wedge z' = z$
OFF, $x' = x \wedge z' = z$
OFF, $x' = x \wedge z' = z$
$x = 20 \wedge z \leq \frac{1}{10}$
UP95, $x \geq 95 \wedge x' = x \wedge x = \frac{1}{10} \wedge z' = 0 \wedge x' = x$
$x = 20 \wedge z = 0$
$\dot{x} = 0 \wedge \dot{z} = 1$
C, $x = 20 \wedge x' = x \wedge z' = z$
$\dot{x} = -Kx \wedge \dot{z} = 1$
$z \leq \frac{1}{10} \wedge 20 \leq x \leq 100$
$\{t_4, th_1\}$
$\{t_3, th_1\}$
DW93, $x \leq 93 \wedge z = \frac{1}{10} \wedge z' = z \wedge x' = x$
DW93, $z = \frac{1}{10} \wedge z' = 0 \wedge x' = x$
ϵ, $93 < x < 95 \wedge z = \frac{1}{10} \wedge z' = 0 \wedge x' = x$
$y = 0$
b_1
$\dot{y} = 0$
TURN-ON, $y' = 0$
$y \leq \frac{1}{10}$
$\dot{y} = 1$
b_2
OFF, $y' = 0$
ON, $y' = 0$
b_4
$\dot{y} = 1$
TURN-OFF, $y' = 0$
$\dot{y} = 0$
b_3
$y \leq \frac{1}{10}$
Burner
$z = \frac{1}{10} \wedge x \geq 95 \wedge z' = 0$, UP95
$z = 0$
$z \leq \frac{1}{10}$
$\dot{z} = 1$
ϵ, $z = \frac{1}{10} \wedge 93 < x < 95 \wedge z' = 0$
$z = \frac{1}{10} \wedge x \leq 93 \wedge z' = 0$, DW93
Thermometer

satisfies ?

$$\Box(\text{low} \leq x \leq \text{high})$$

System

Math. model

Model Based Development

❶ Make a model of the environment
Env

❷ Make a model of your control strategy
Controller

❸ Make clear the control objective
(avoid) **Bad**

❹ Verify that
Does **Env** || **Controller**
avoid **Bad** ?

Model Based Development

❶ Make a model of the environment
Env

❷ Make a model of your control strategy
Controller

❸ Make clear the control objective
(avoid) **Bad**

Good but after ?

roller
avoid **Bad** ?

Model Based Development

❶ Make a model of the environment
Env

❷ Make a model of your control strategy
Controller

❸ Make clear the control objective
(avoid) **Bad**

Good but after ?

We want correct implementations !

From correct models to correct implementations

Should we verify code ?

-- This may be too difficult (too much details)

Translate models into code ?

-- There are tools for that (Simulink)

From correct models to correct implementations

Should we verify code ?

-- This may be too difficult (too much details)

Translate models into code ?

-- There are tools for that (Simulink)

... and preserve good properties ?

-- Good question...

From correct models to correct implementations

Should we verify code ?

-- This may be too difficu

Translate models in

-- There are tools for that

... and preserve goo

-- Good question...

Unfortunately, timed automata are (in general) **not** *implementable* :

Zenoness: 0, 0.5, 0.75, 0.875, ...

No minimal bound between two transitions: 0,0.5,1,1.75,2,2.875,3,...

And more: instantaneity, real-valued clocks... (robustness)

From correct models
to correct implementations

Should we verify code ?

-- This may be too difficu

Translate models in

-- There are tools for that

Unfortunately, timed automata are (in general) **not** *implementable* :

Zenoness: 0, 0.5, 0.75, 0.875, ...

No minimal bound between two transitions: 0,0.5,1,1.75,2,2.875,3,...

nstantaneity, real-valued clocks...

What if my control strategy is "correct" for one of those reasons ?

A solution: Almost ASAP semantics

Alternative semantics for timed automata

❶ Enabled transitions of the controller become urgent after Δ time units;

❷ Events from the environment are received by the controller within Δ time units;

❸ Truth values of guards are elarged by f(Δ)

A solution: Almost ASAP semantics

Alternative semantics for timed automata

❶ Enabled transitions of the controller become urgent after Δ time units;

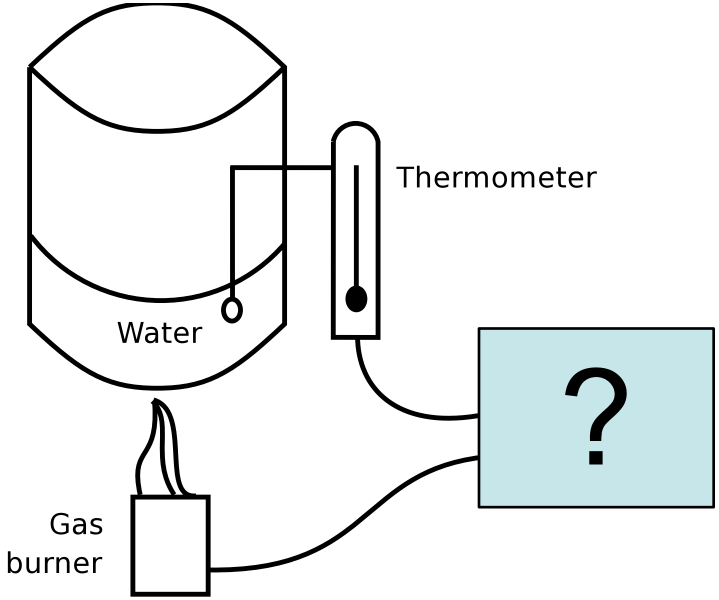
❷ Events from the environment are received by the controller within Δ time units;

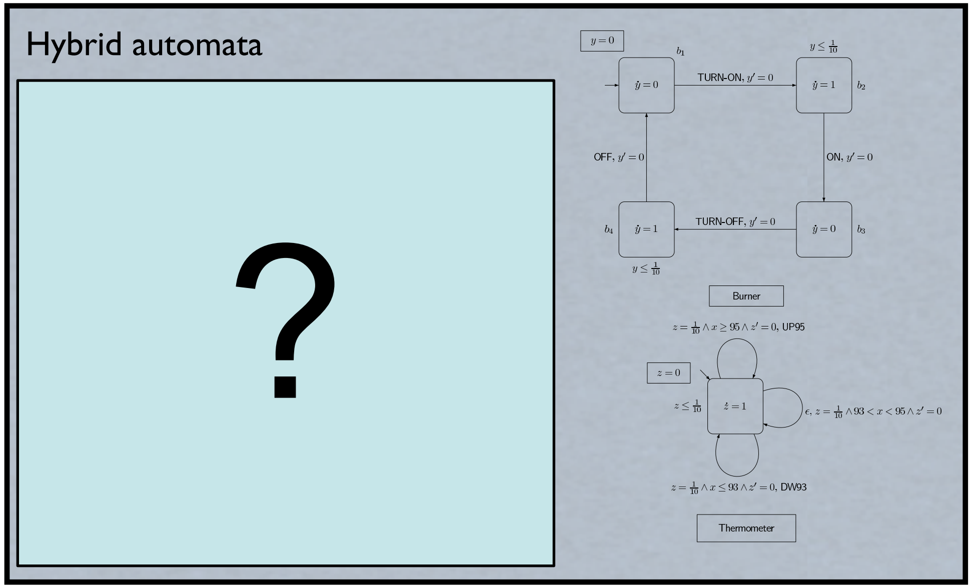
❸ Truth values of guards are elarged by f(Δ)

✓ **AASAP semantics is implementable**

✓ **Prototypes of tools to verify AASAP semantics and generate provably correct code have been implemented**

Model Based ~~Development~~ Synthesis





satisfies ?

$$\Box(\text{low} \leq x \leq \text{high})$$

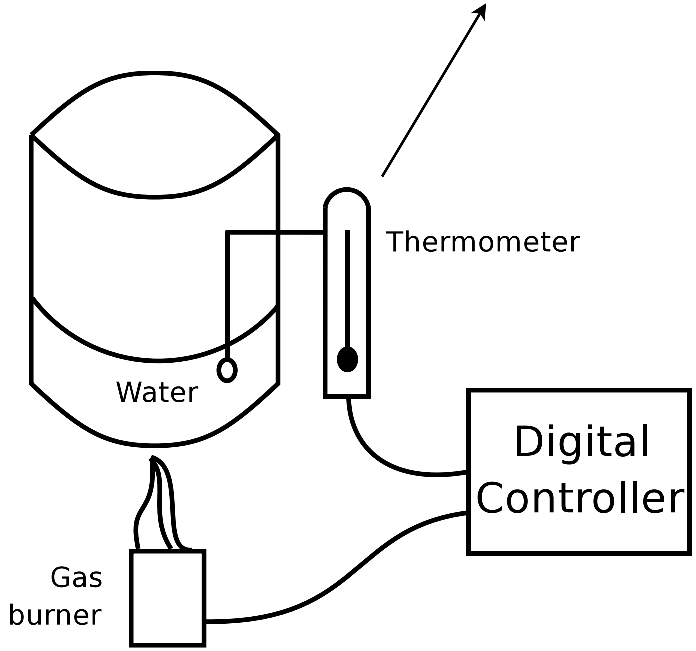
System

Math. model

Classical algorithms for synthesis: *perfect information hypothesis*

Finite precision = imperfect information

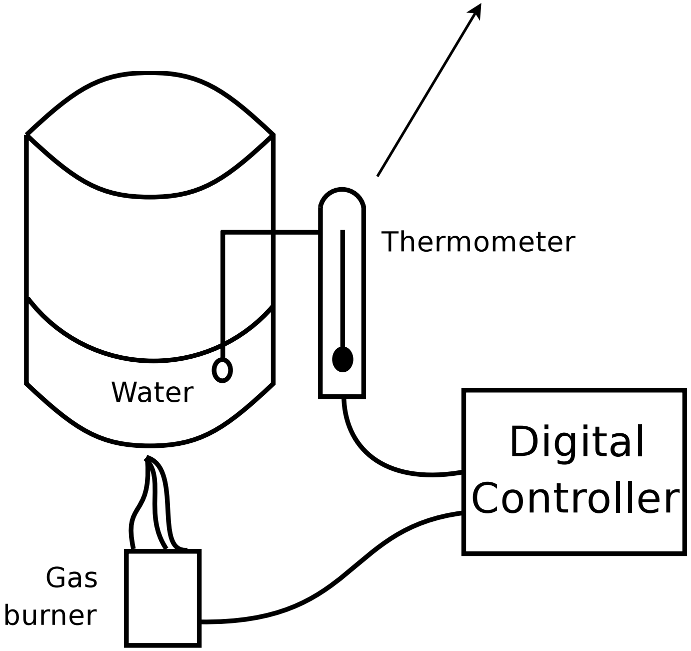
The temperature is in $(C\text{-}I, C\text{+}I)$



Classical algorithms for synthesis: *perfect information hypothesis*

Finite precision = imperfect information

The temperature is in (C-l,C+l)



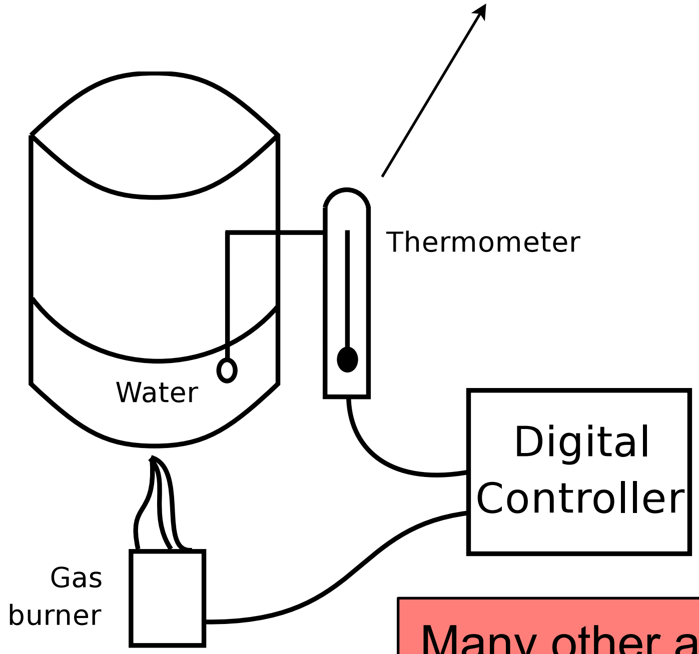
For **robust controllers**: we must drop the perfect information hypothesis!

We propose **new algorithms** to synthesize **observation based strategies**:
we **avoid determinization** !

Classical algorithms for synthesis: *perfect information hypothesis*

Finite precision = imperfect information

The temperature is in (C-I,C+I)



For **robust controllers**: we must drop the perfect information hypothesis!

We propose **new algorithms** to synthesize **observation based strategies**:
we **avoid determinization** !

Many other applications of the idea are forseen: e.g. **improved algorithms for the automata based approach to model-checking.**

Recent publications

- Khrishnendu Charterjee, Laurent Doyen, Thomas A. Henzinger and Jean-Francois Raskin. **Algorithms for Omega-regular games of Incomplete Information**. To appear in *CSL'06,* Lecture Notes in Computer Science, 2006. (16 pages)
- Martin De Wulf, Laurent Doyen, Thomas A. Henzinger and Jean-Francois Raskin. **Antichains: a New Algorithm to Solve Universality of FA**. In *CAV'06*, Lecture Notes in Computer Science, 4144, Springer-Verlag, pp. 17-30, 2006.
- Martin De Wulf, Laurent Doyen, and Jean-Francois Raskin. **A Lattice Theory for Solving Games of Imperfect Information**. In *HSCC'06*, Lecture Notes in Computer Science, 3927, pp. 153-168, Springer-Verlag, 2006.
- Martin De Wulf, Laurent Doyen, Jean-François Raskin. **Systematic Implementations of Timed Models**. In *FM'05*, Lecture Notes in Computer Science 3582, pp. 139--156, Springer-Verlag, 2005.
- Martin De Wulf, Laurent Doyen, Jean-François Raskin. **Almost ASAP Semantics: from Timed Models to Timed Implementations**.In *Formal Aspect of Computing*, 17(3):319--341, Springer-Verlag, 2005.
- Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. **Robustness and Implementability of Timed Automata**. In *FORMATS'04*, Lecture Notes in Computer Science, 3253, pp. 118-133, Springer Verlag, 2004.
- Martin De Wulf, Laurent Doyen, Jean-François Raskin. **Almost ASAP Semantics: From Timed Models to Timed Implementations**. In *HSCC'04*, Lecture Notes in Computer Science, 2993, pp 296-310, 2004.
- Tech. Rep. 2006.76: Laurent Doyen (ULB), Jean-François Raskin (ULB), **Improved Algorithms for the Automata-Based Approach to Model-Checking**. Submitted. 2006.