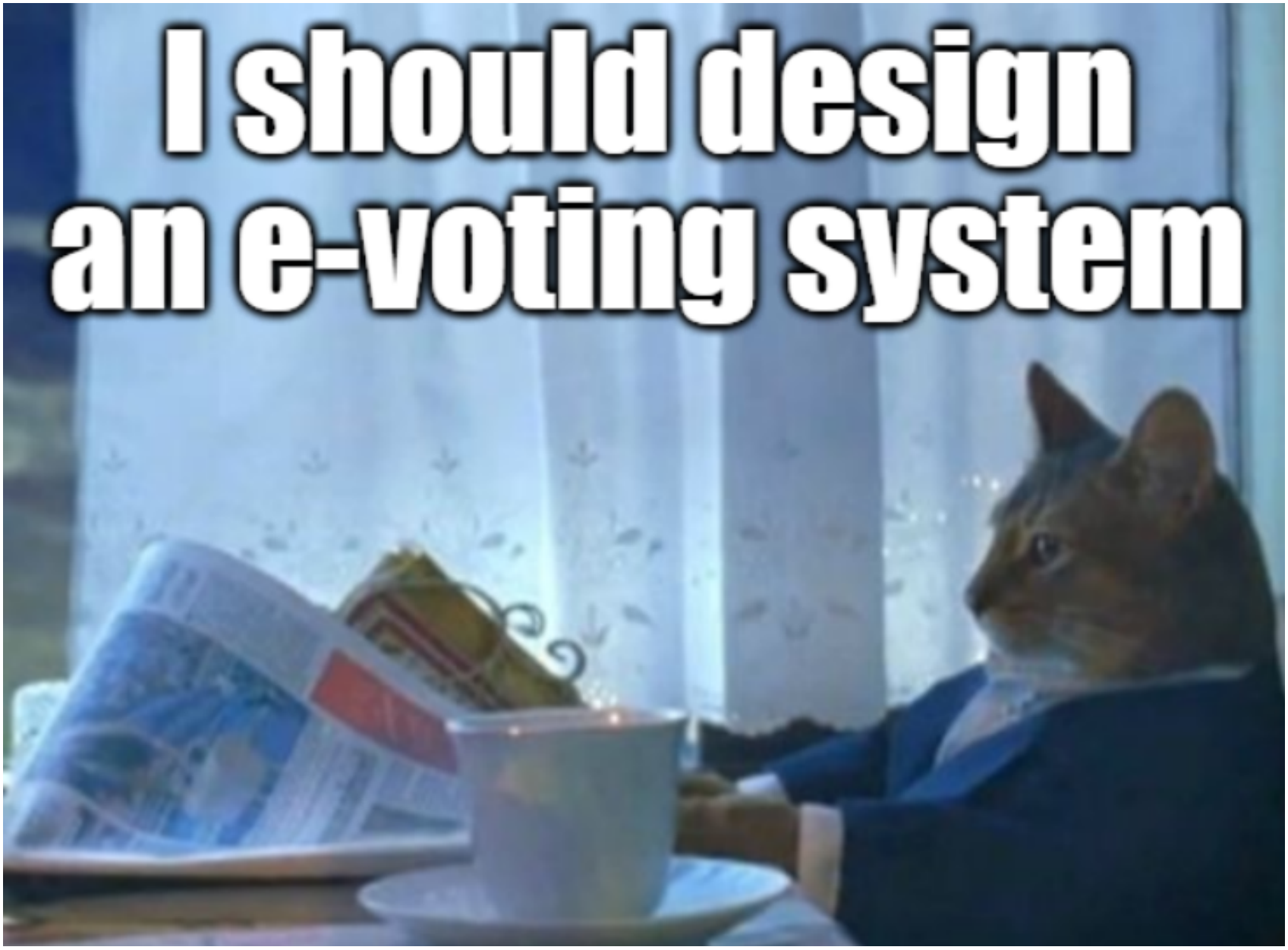
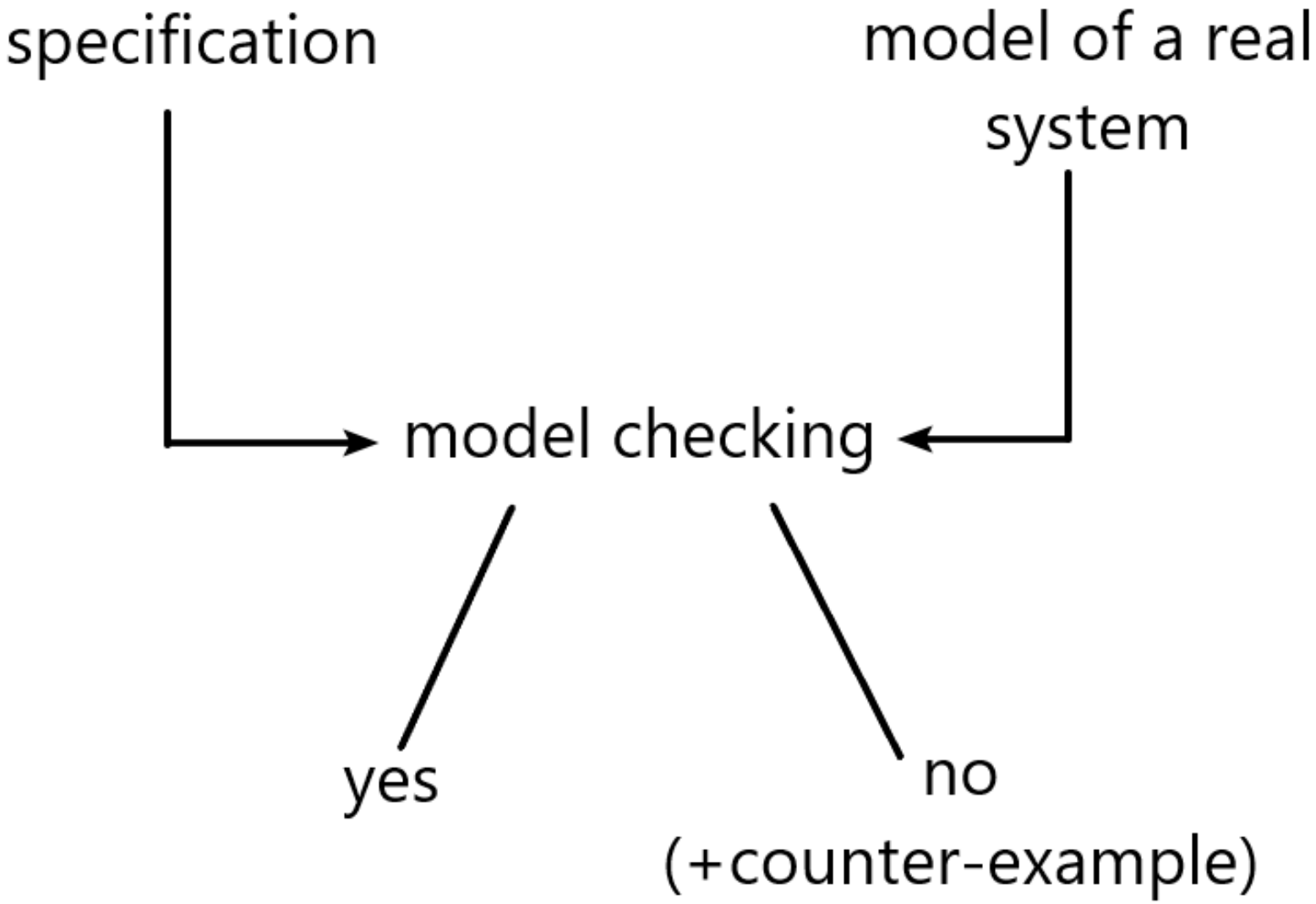
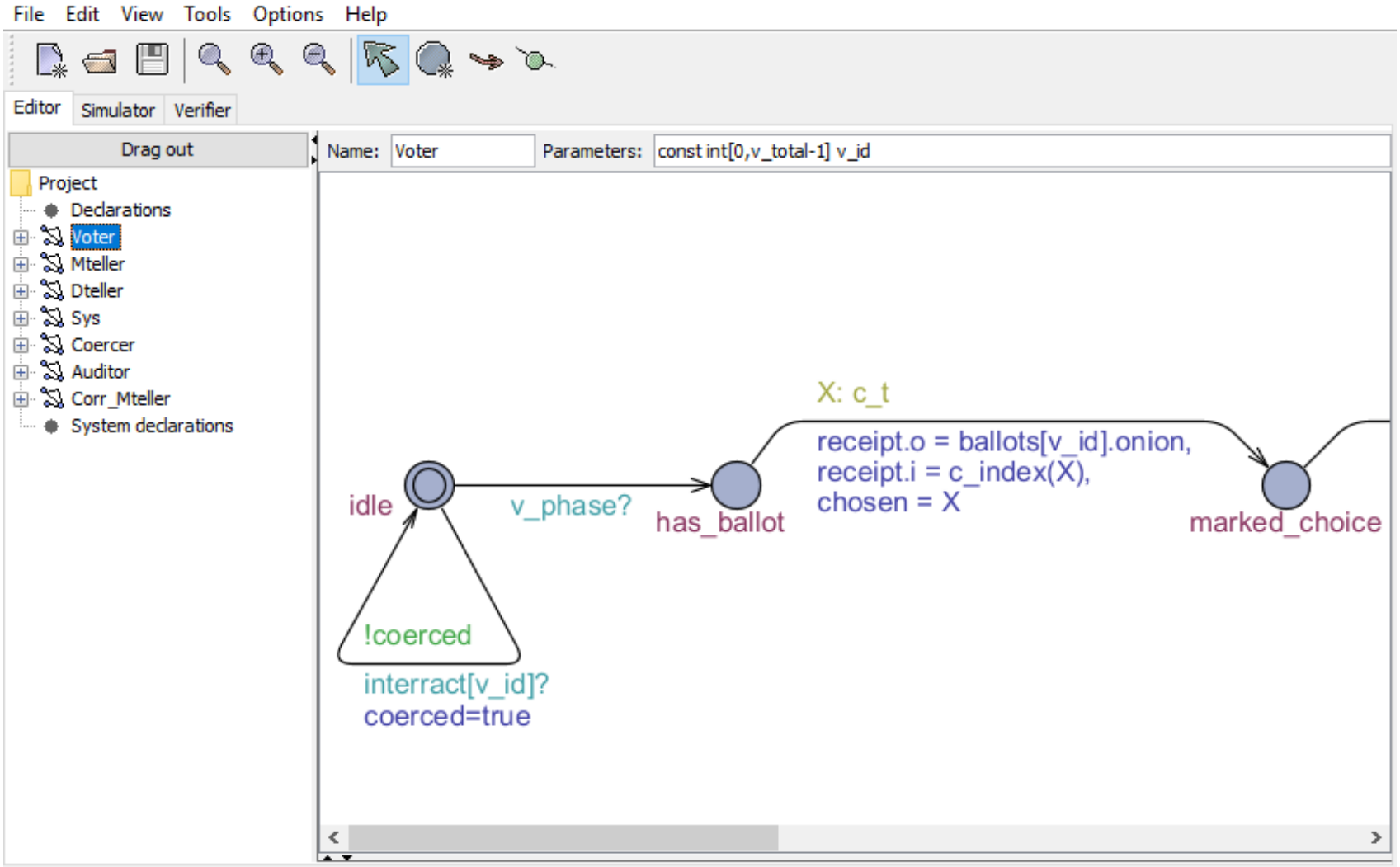
Towards Model Checking of Voting Protocols in Uppaal

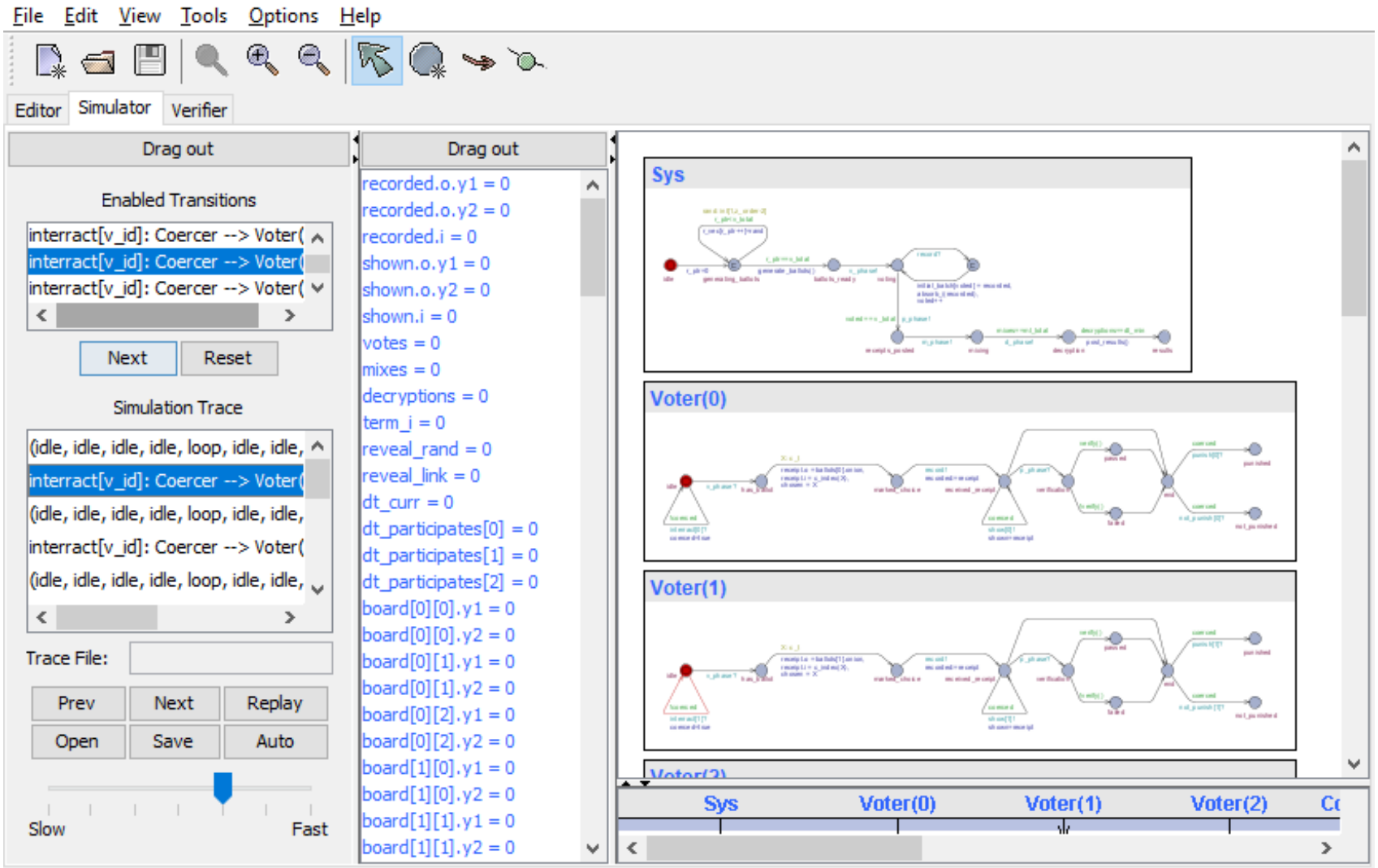
Wojciech Jamroga, Peter Y. A. Ryan,
Damian Kurpiewski, Yan Kim

October 1, 2020


I should design
an e-voting system


specification
model of a real system
model checking
yes
no
(+counter-example)


File Edit View Tools Options Help
Editor Simulator Verifier
Drag out
Project
Declarations
Voter
Mteller
Dteller
Sys
Coercer
Auditor
Corr_Mteller
System declarations
Name: Voter
Parameters: const int[0,v_total-1] v_id
idle
v_phase?
has_ballot
X: c_t
receipt.o = ballots[v_id].onion,
receipt.i = c_index(X),
chosen = X
marked_choice
!coerced
interract[v_id]?
coerced=true


File Edit View Tools Options Help
Editor Simulator Verifier
Drag out
Enabled Transitions
interract[v_id]: Coercer --> Voter(
interract[v_id]: Coercer --> Voter(
interract[v_id]: Coercer --> Voter(
Next Reset
Simulation Trace
(idle, idle, idle, idle, loop, idle, idle,
interract[v_id]: Coercer --> Voter(
(idle, idle, idle, idle, loop, idle, idle,
interract[v_id]: Coercer --> Voter(
(idle, idle, idle, idle, loop, idle, idle,
Trace File:
Prev Next Replay
Open Save Auto
Slow Fast
Drag out
recorded.o.y1 = 0
recorded.o.y2 = 0
recorded.i = 0
shown.o.y1 = 0
shown.o.y2 = 0
shown.i = 0
votes = 0
mixes = 0
decryptions = 0
term_i = 0
reveal_rand = 0
reveal_link = 0
dt_curr = 0
dt_participates[0] = 0
dt_participates[1] = 0
dt_participates[2] = 0
board[0][0].y1 = 0
board[0][0].y2 = 0
board[0][1].y1 = 0
board[0][1].y2 = 0
board[0][2].y1 = 0
board[0][2].y2 = 0
board[1][0].y1 = 0
board[1][0].y2 = 0
board[1][1].y1 = 0
board[1][1].y2 = 0
Sys
Voter(0)
Voter(1)
Sys Voter(0) Voter(1) Voter(2)

Modelling in Uppaal

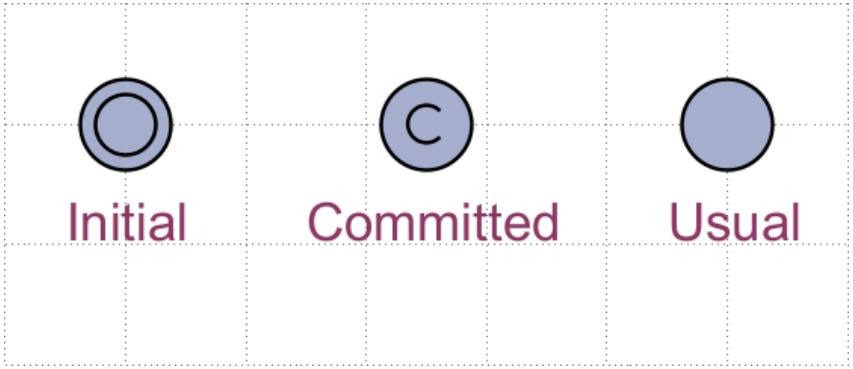
An Uppaal model is a set of concurrent *processes*.
Processes are instantiated of *templates*, each possibly having a list of parameters.
Templates allow to define a large number of almost identical processes.

Nodes (locations)

Depicted by circles and represent a local state of module.

Can be annotated by name label (unique within a template).

Initial nodes are marked by *double circle*.
Committed nodes are marked by *circled C*.

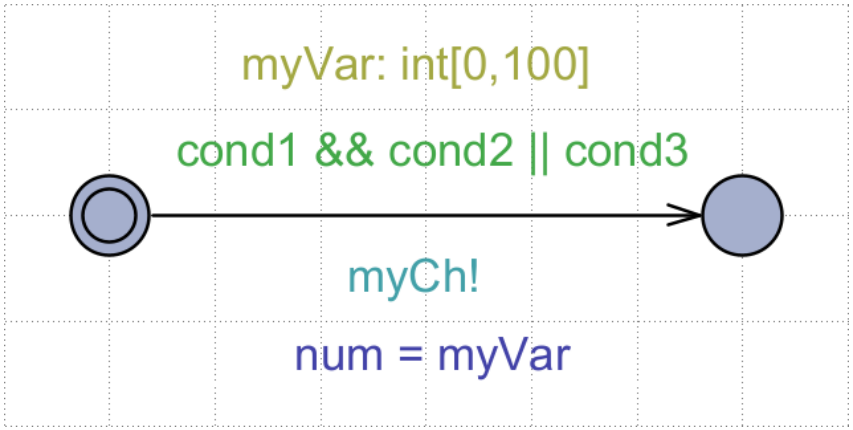


Edges

Define the local transitions in the module.

Can be annotated by:

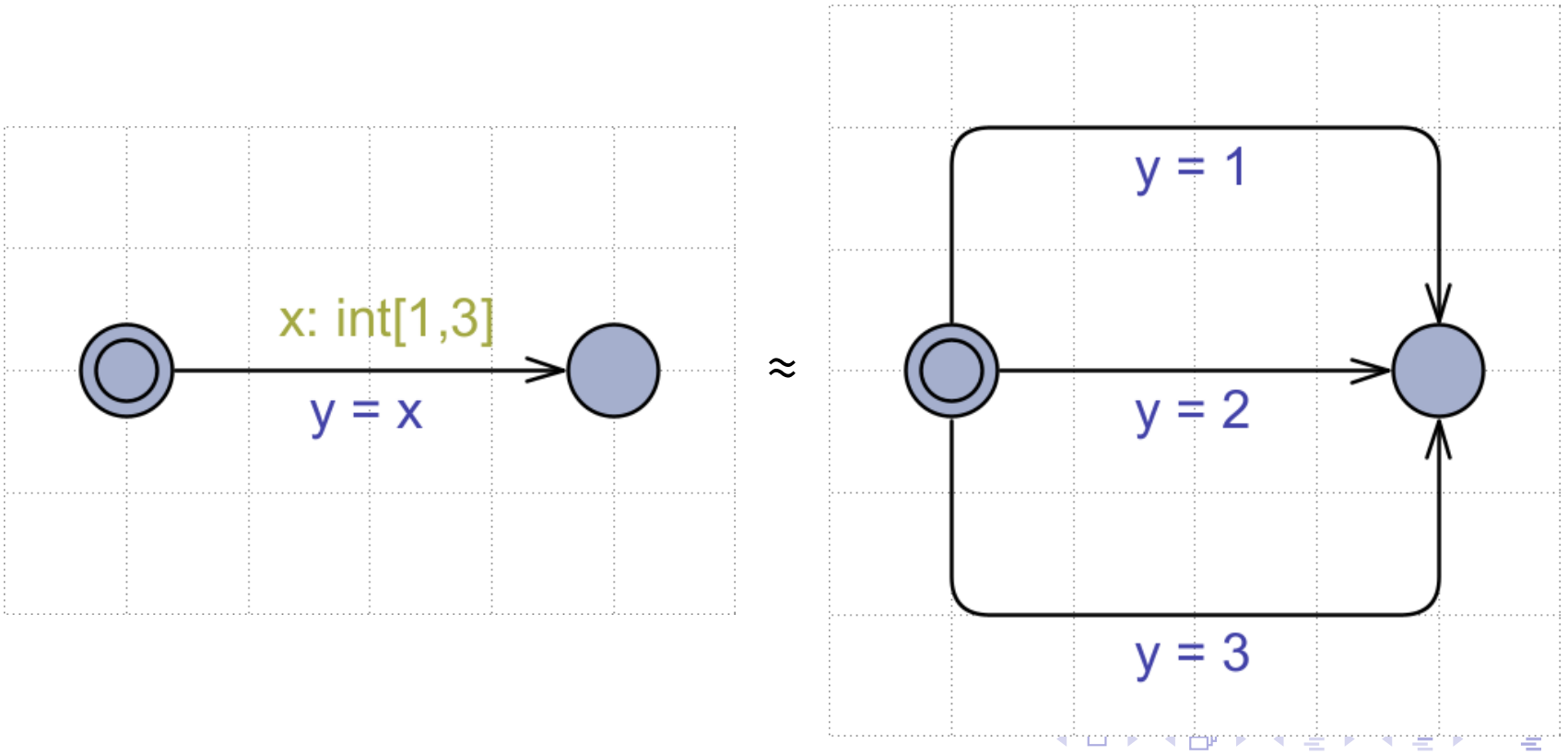
- selection
- guard
- synchronization
- update



Selection&Update example

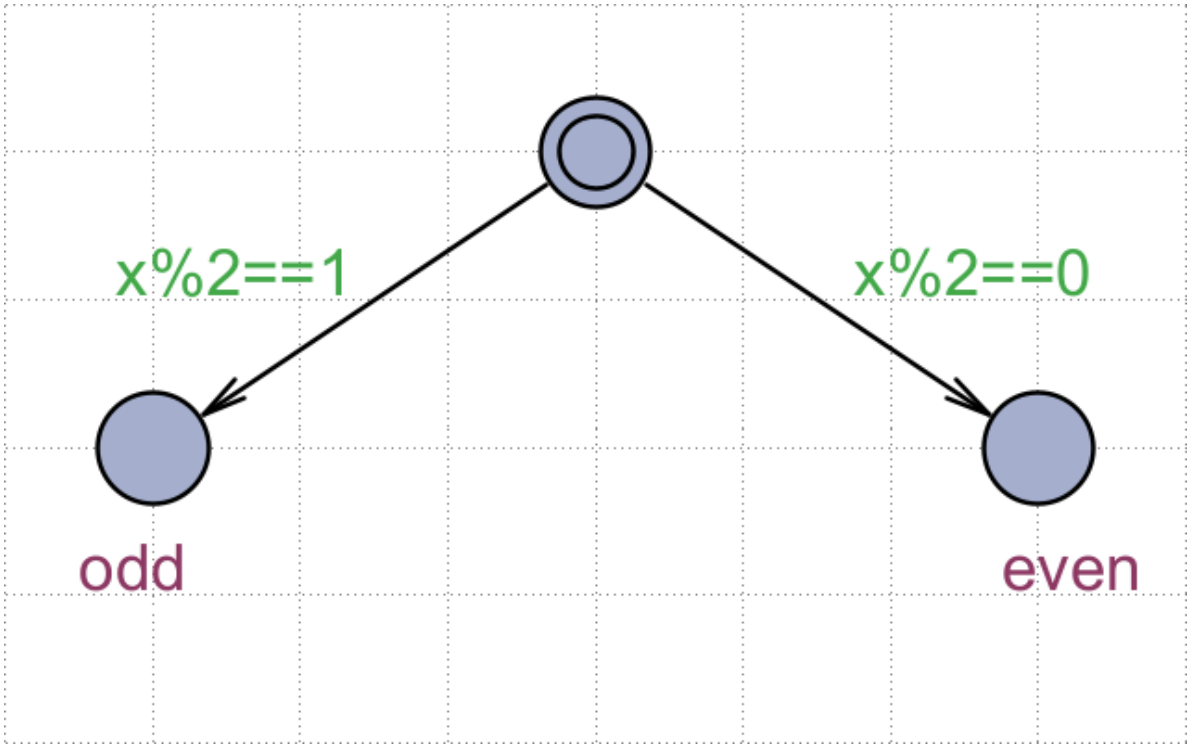
Selection binds the identifier (on the left) to a value from a given range (on the right) in a non-deterministic way.

Update expression is evaluated when transition is taken.



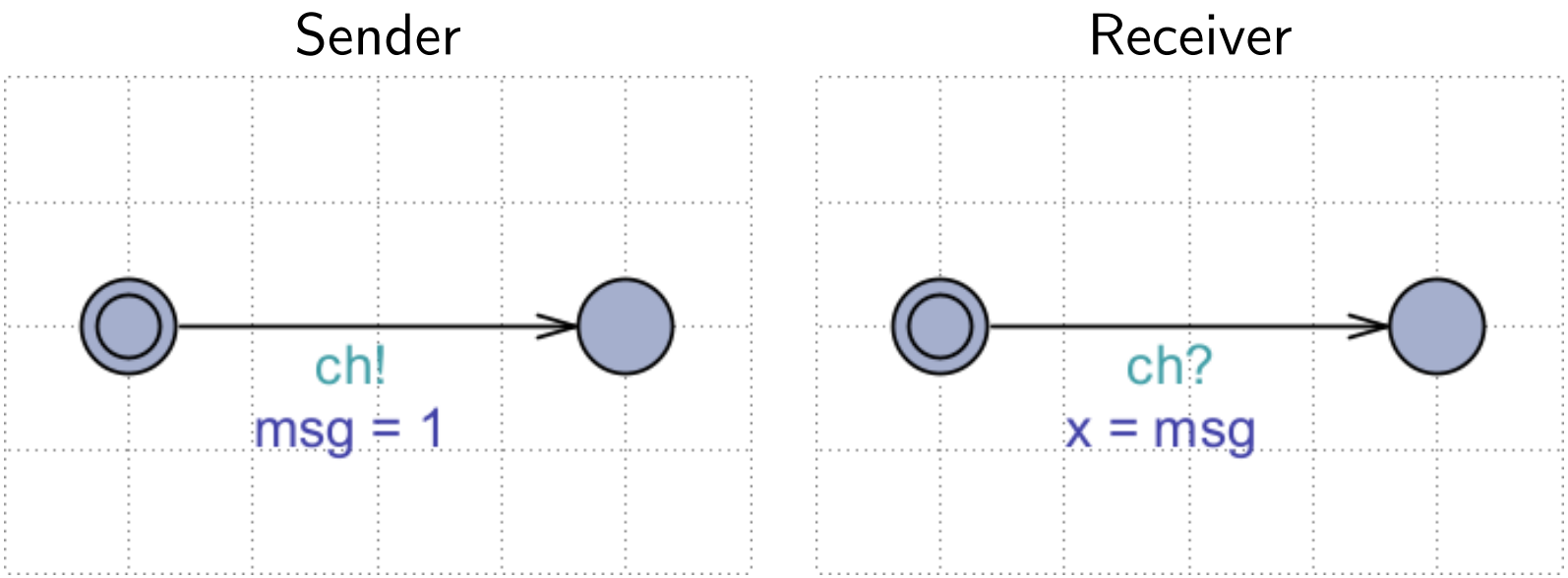
Guard example

Guard enables the the transition if and only if the guard condition evaluates to `True`.



Synchronization example

Synchronization allows two or more processes to synchronize over a common channel.
In order to pass value over a channel, one has to use shared global variables for transmission.



Specification syntax

Supported properties:

- Possibly: `E<>p`
- Invariantly: `A[]p`
- Potentially always: `E[]p`
- Eventually: `A<>p`
- Leads To: `p-->q` (= `A[](p imply A<>q)`)

Voter verifiable system

Common scenario:

- at the time of casting an encr./enc. of the vote is created and posted to a secure public BB
- voter can later check that her encr. ballot appears correctly
- set of posted ballots are then processed (in some verifiable way) to reveal the tally or outcome

Voter verifiable system

Common scenario:

- at the time of casting an encr./enc. of the vote is created and posted to a secure public BB
- voter can later check that her encr. ballot appears correctly
- set of posted ballots are then processed (in some verifiable way) to reveal the tally or outcome

Challenge:

- assurance of the accurate outcome
- avoid introducing any coercion threats

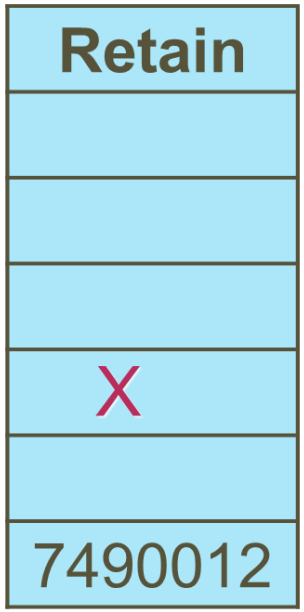
Ballot form

Destroy	Retain
Asterix	
Obelix	
Idefix	
Panoramix	
Abraroucourix	
	7490012

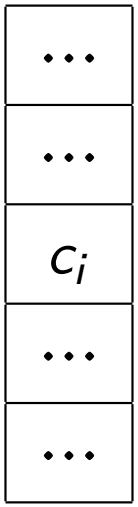
Ballot form

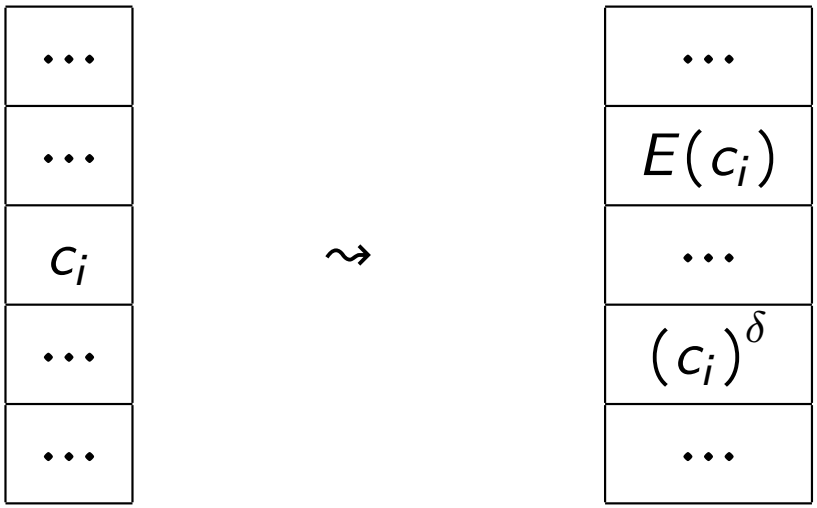
Destroy	Retain
Asterix	
Obelix	
Idefix	
Panoramix	X
Abraroucourix	
	7490012

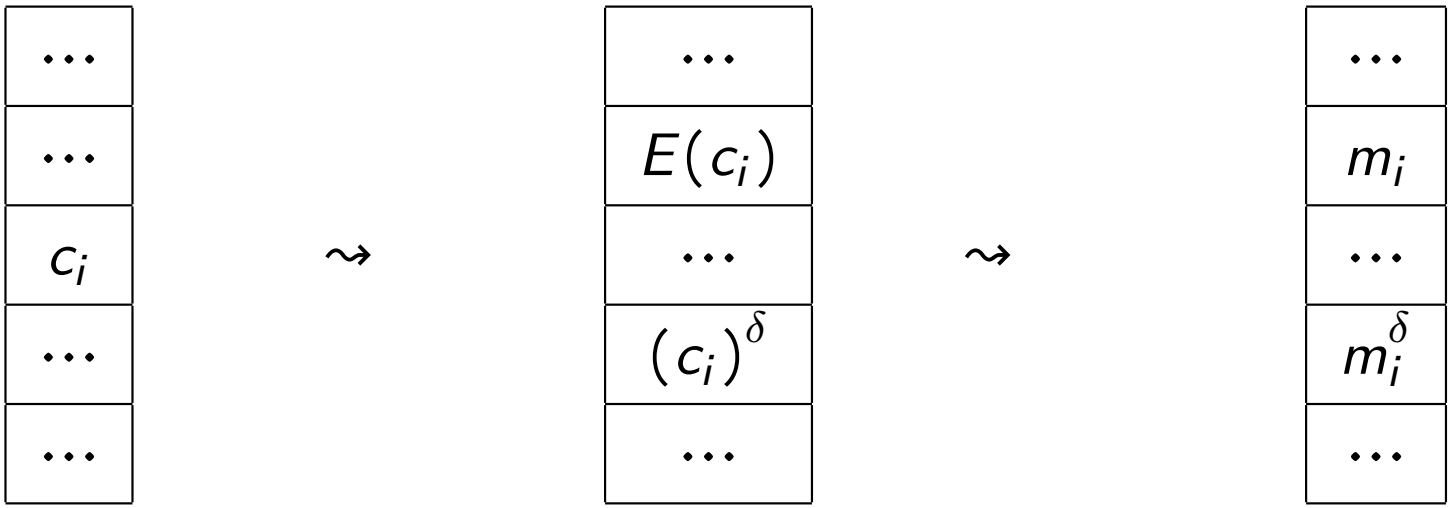
Ballot form



Presentation of the model in Uppaal


...
...
c_i
...
...


...
...
c_i
...
...
$\rightsquigarrow$
...
$E(c_i)$
...
$(c_i)^{\delta}$
...


...
...
c_i
...
...
⇝
...
$E(c_i)$
...
$(c_i)^{\delta}$
...
⇝
...
m_i
...
m_i^{δ}
...

Thank You for Your time!
Q&A